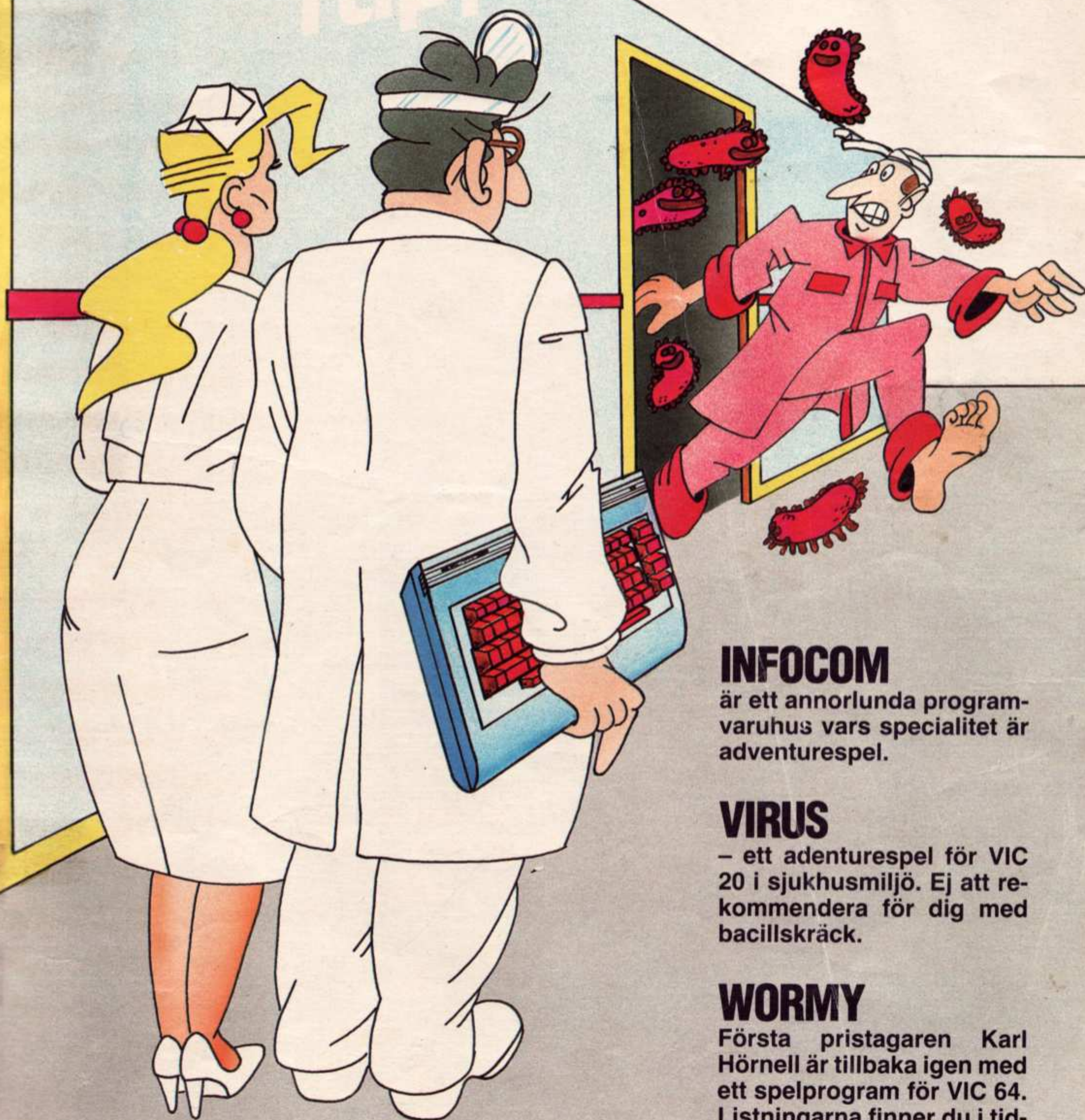
VIC

1985 NR 2

PRIS 17:50 inkl moms



INFOCOM
är ett annorlunda programvaruhus vars specialitet är adventurespel.

VIRUS
– ett adenturespel för VIC 20 i sjukhusmiljö. Ej att rekommendera för dig med bacillskräck.

WORMY
Första pristagaren Karl Hörnell är tillbaka igen med ett spelprogram för VIC 64. Listningarna finner du i tidningen.

473–2

VIC rapport

INNEHÅLL NR. 2 1985

VIC redaktion: Box 42054, 126 12 Stockholm. Telefon 08-744 59 20
Ansvarig utgivare: Nina Linander
Administrativ redaktör: Elisabeth Höglund
Teknisk redaktör: Matts Nilsson
Omslagsteckning: Per Lindforss
Layout: Ordbyggarna
Övriga medarbetare: Åke Hedman, Joakim Aspengren, Ola Johansson, Åke Fredriksson, Bengt Litnäs, Ola Axelsson, Thomas Wernersson.
Annonser: Marknadsmedia AB. Telefon 08-753 00 20
Tryck: Per·Bo AB, Stockholm 1985
ISSN-NR: 0281–8043
Annonsorder och annonsmaterial (hel- original eller negativ film) enligt överenskommelse.
Tryckförfarande: Offset.
Upplaga 20 000 ex.
Nr 3 1985 av VIC rapport utkommer den 9 april. Manusstopp för nr 4 är den 25 mars och annonsstopp den 10 april.

Äntligen lite action

Piratkopiering har länge varit ett stort problem för programvaruhusen. Dessa lägger ner stora summor på programutveckling som de väntar sig att få tillbaka via framtida försäljning. Men när programmet släpps ut till allmänheten finns det några "smarta" programmerare som bryter kopieringsskyddet och sedan tjänar pengar själva. Denna verksamhet har knäckt många programvaruföretag och 1984 fick vi se en del av dessa i konkurs.

Runt om i världen märker man nu en reaktion. I flera länder har piratkopierare ertappats och fällts i domstol. Prejudikat dyker upp i land efter land.

I Sverige har programvaruföretagen blivit mer aktiva. Efter ett känt fall inom videobranschen där svartkopierarna fälldes har programvaruföretagen aktiverats.

Personligen önskar jag dem lycka till i jakten på piratkopierare så att vi även i framtiden kan njuta av väl förberedda och avancerade datorprogram.

Slut på lagret

VIC rapport årg 3 nr 8 och 9 samt nr 1/85 är slut på lagret. Går ej att beställa.

COMMODORE color MCS 801 graphic printer

Fördelarna med denna printer. Den har åtta färger som är enkla att hantera, man kan skriva i olika storlekar, olika radavstånd, reverserat, skriva samma text två gånger på samma ställe så kallat "2-pass mode" och mycket mera.

En av de största fördelarna som jag ser det är att papperet inte trasslar sig. Konstruktionen är simpel men effektiv. Det är en platta som skiljer det undre papperet från det övre och en som styr det övre papperet så att det inte kommer snett eller tillbaks under valsen.

Här är en liten demonstration av färghantering.

```
10 OPEN4,4
20 FOR T=1 TO 8
30 PRINT#4,CHR$(20),CHR$(T),
"COMMODORE"
40 NEXT
50 CLOSE4
```

Så här kommer man in i "2-pass mode"

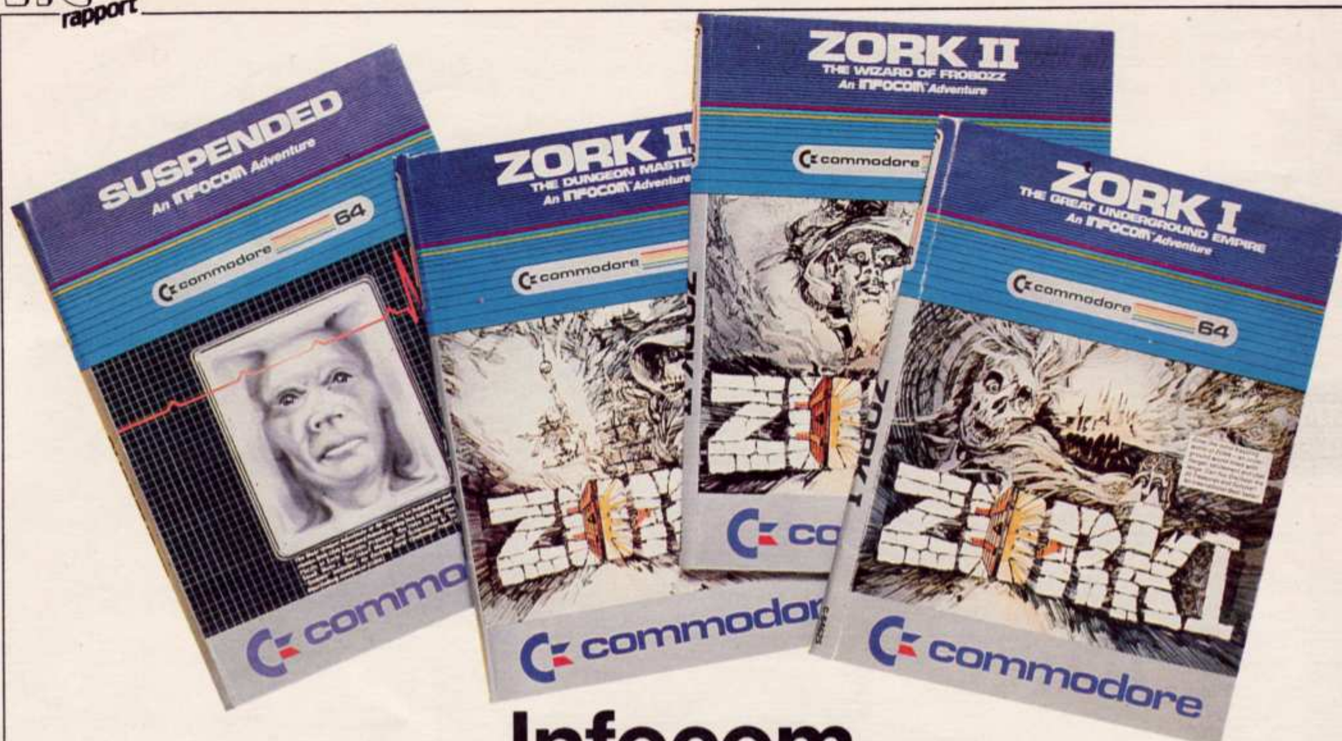
```
10 OPEN19,4,19:PRINT#19:CLOSE19
```

Det finns speciellt en sak som är rolig för den som är intresserad av att konstruera egna tecken direkt till printern. Så här kan det se ut om man vill göra sig en liten gubbe.

```
10 DATA 0,16,203,63,203,16,0,0
20 OPEN 5,4,5
30 FOR T=1 TO 8:READ A:A$=A$+CHR$(A-
):NEXT
40 OPEN7,4,7
50 DC$=CHR$(20): REM CHR$(20) sätts innan
man byter färg
60 PC$=CHR$(254): REM CHR$(254) står för det
nya tecknet
70 CS$=CHR(141): REM CHR$(141) står för
carrige return
80 EN$=CHR$(14): REM CHR$(14) sätter extra
stora bokstäver
90 DE$=CHR$(15): REM CHR$(15) stänger av
extra stora bokstäver
100 PRINT#7,DC$;CHR$(3)
110 PRINT#5,A$:PRINT #7,EN$;" ";PC$;CS$
120 END
```

NACKDELARNA MED MCS 801

Den skriver grått i stället för svart, vilket gör att det ibland blir svårt att se vad det står. För att motverka detta måste man gå in i "2-pass mode". Det gör att den blir långsammare något tråkigt att använda. De andra färgerna t ex rött och blått syns klart bättre. Det är svårt att sätta dit färgbandet, p g a att det är trångt mellan valsen och den lilla stålbrickan som centrerar färgbandet. Det är också svårt att sätta dit papperet, som också hade en tendens att fastna på samma ställe som färgbandet. □



Infocom

Bland alla spel som finns på marknaden idag till Commodore 64, är det få som är så genomtänkt gjorda som ett riktigt bra adventure. Många av de människor som tycker illa om grafikactionspelens pang–svisch–egenskaper, lutar sig mycket hellre tillbaka med ett gott adventure i sin 64. Det kan gälla att rädda en stad, eller att hitta mördaren i ett gammalt klassiskt deckaräventyr. Adventurespelandet skänker avkoppling men kräver också 100%-ig tankeskärpa. De kan ta veckor att lösa, även om man spenderar 5 timmar per natt på att kämpa sig fram.

Olika adventures är olika komplext uppbyggda. De kan vara enkla att lösa, eller mer komplicerade. Detta beror främst på storleken hos spelet. Ett litet textadventure kan vara på 8k, medan ett stort grafikdito kan ta 10 disketter i anspråk. Grafikbilderna tar otroligt mycket plats, så av ett grafikadventure och ett textdito av samma klass, är grafikspelet kanske 15 g g r så stort.

Själv föredrar jag riktigt stora och mystiska textadventures. Där kan man få stora spel med bra 'story', utan att det tar 5 dygn att ladda in. Det verkar dessutom som om en del grafikspelsskapare tänker mer på bilderna, och glömmer bort själva spelet litegrann. De flesta grafikspel håller inte alls samma klass som de bästa textspelen.

De absolut bästa adventurespelen idag tillverkas av det amerikanska företaget Infocom Inc. De har gett ut en serie adventures till 64:an, (även PC, Apple m m). Samtliga spel är på disk och alla är helt textbaserade. De är vardera ungefär 80–160k stora, och har i och med detta ett väl tilltaget vokabulär och en rymlig värld att vandra runt i. De flesta av Infocoms adventurespel levereras med en massa tillbehör till. Det kan vara ledtrådar till deckaräventyret, eller en karta över området man jobbar med. Detta har gjort att man förutom att bara behandla text från datorn, också får lite andra saker att plocka med. Dessutom har det gjort spelen betydligt mindre populära att kopiera.

Infocom har gett ut en hel serie spel, som alla håller mycket hög kvalitet. De namn som hörts hittills är:

- **Deadline**
- **Suspended**
- **Zork I, II och III**
- **Starcross**
- **Witness**
- **Planetfall**
- **Infidel**
- **Enchanter**
- **Sorcerer**

En hel del av dessa finns inte på den svenska marknaden riktigt än, men dyker upp så småningom, (får man hoppas). Att berätta mer om alla skulle vara alldeles för jobbigt, så jag ska senare plocka ut ett par av dem, och förklara lite närmare vad de går ut på. Men först ett kort stycke allmänt om dem.

PROGRAM-RECESION

Alla spel är alltså helt uppbyggda av text. De har ett enormt vokabulär, vilket gör att man kan konversera med datorn på i stort sett vanlig ren engelska. Den förstår till och med flerledade kommandon som 'take the book and then go north', eller 'examine all but the sack and the newspaper'.

Riktningar kan ges i samma form som de flesta adventures, d v s 'N', 'S', 'W', och 'E'. Dessutom kan man be datorn själv tänka efter vart man ska gå, t ex genom att skriva 'walk around the house'.

Det kan löna sig att sitta ner och prova vilka kommandon den förstår. Detta brukar dessutom vara rätt kul. Prova t ex vad Zork svarar när man skriver 'jump', eller 'eat mailbox'...

Förutom de vanliga orden har Infocoms adventures en del specialfunktioner. Man kan förstås spara sitt spelande på disk, (för att hinna få någon timmes sömn innan jobbet börjar igen), och sedan ladda in det igen och fortsätta. Man kan få ut sitt spelande på skrivare, om man har en sån ansluten. Dessutom har alla spel en mängd specialkommandon, som t ex 'configure' i Suspended, eller 'diagnose' i Zork. Dessa finns uppräknade i respektive spels instruktionsbok.

Som exempel på dialogen i spelen har jag tagit med en bit från en match mot Zork I. Jag har kommit in i en grottsal, där jag träffar på ett troll.

ZORK

Det mesta kända av Infocoms spel är trilogin Zork. Det har blivit så populärt att det finns allt från böcker till bilbakrutesdekaler att köpa om spelet, (i USA förstås). Det lär till och med finnas speciella Zork-klubbar där borta i väster... Zork utspelar sig i underjorden. Man börjar dock sitt spelande ovan jord, så för den ovane tar det en del tid bara att hitta ner till rätt nivå. Man ska sedan hitta skatter, och på så sätt skaffa sig poäng.

DEADLINE

Deadline är det första deckaradventuret. En man har mördats, och du har 12 timmar på dig att lösa mysteriet. Till detta spel medföljer en låda med ledtrådar, som en liten pillerburk, en trave undersökningsprotokoll med fingeravtryckstester, m m m m. Du får gå runt brottsplatsen med omgivningar och förhöra folk, undersöka fotspår, och lyssna på intressanta telefonsamtal.

SUSPENDED

Om du tycker att du börjar bli skärpt på adventurespelande, eller bara är smart i allmänhet, ska du ge dig på Suspended. Det är det klart mest komplexa av spelen, och lite knepigare att knäcka än de andra. Du har fått ta över styret över en stad, vars förre ledare tyvärr blev sinnessjuk och började mörda befolkningen. Han lyckades dessutom förstöra en del av den tekniska utrustning som skulle ha funnits till din hjälp, så du kan stöta på vissa problem i din uppgift...

Med Suspended följer en karta över 'The Underground Complex', vilket blir ditt hem de närmaste 500 åren.

Sammanfattningsvis kan jag lugnt rekommendera alla Infocoms kreationer till alla vänner av logiskt tänkande och äventyr. De är det bästa som går att få tag på i adventureväg idag, och kommer att ge dig många trevliga dygn av rikligt kaffedrickande och total sömnbrist. Att lösa ett av dessa spel ger dig inte en kvarts miljon kronor i handen, men däremot den tillfredsställelse som bara ett riktigt bra adventure kan ge. □

Måns Johnsson, WVK

Program "Zork 1"

>LIGHT LANTERN
The lamp is now on.
Cellar
You are in a dark and damp celler with a narrow passageway leading north, and a crawlway to the south. On the west is the bottom of a steep metal ramp which is unclimbable.

>N
The Troll Room
This is a small room with passages to the east and south and a forbidding hole leading west. Bloodstains and deep scratches (perhaps made by an axe) mar the walls.
A nasty-looking troll, brandishing a bloody axe, blocks all passages out of the room.
Your sword has begun to glow very brightly.

>KILL TROLL
(with the sword)
A quick stroke, but the troll is on guard.
The axe sweeps past as you jump aside.

>AGAIN
The troll is confused and can't fight back.
The troll slowly regains his feet.

>AND AGAIN
The troll is momentarily disoriented and can't fight back.
Someone carrying a large bag is casually leaning against one of the walls here. He does not speak, but it is clear from his aspect that the bag will be taken only over his dead body. The troll slowly regains his feet.

>KILL TROLL
(with the sword)
A good slash, but it misses the troll by a mile.
You stagger back under a hail of axe strokes.

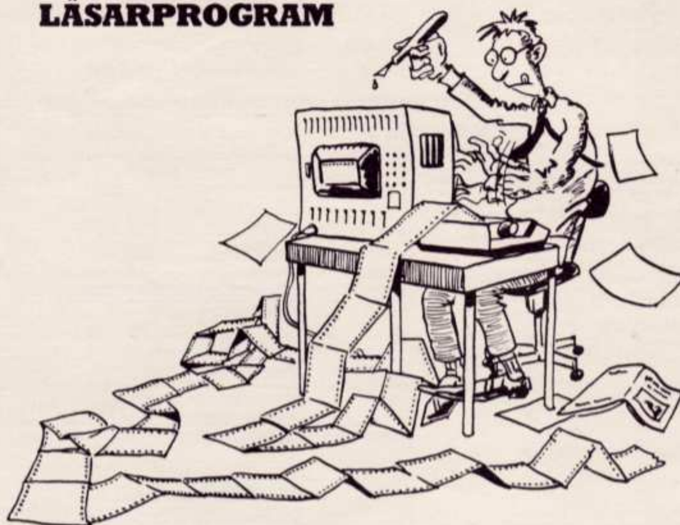
>AGAIN
You are still recovering from that last blow, so your attack is ineffective. The other occupant just left, still carrying his large bag. You may not have noticed that he robbed you blind first.
The axe sweeps past as you jump aside.

Forts på sidan 8

En assemblerkompilator i Basic – VIC 20 –

Ett program till alla dem som vill försöka programmera i en kraftfull ASSEMBLER anpassad till VIC 20 utan att köpa en maskinkodsmonitor.

LÄSARPROGRAM



1. En instruktion heter WOR. (Word) Kompilatorn beräknar operandens värde och lägger in värdet i två bytar. Denna instruktion kan t ex användas om man vill skapa en lista med adresser.
2. Assemblerprogrammet måste avslutas med instruktionen END. (Måste också tillfogas i programexemplen.)

Ett ASSEMBLER-program består precis som ett BASIC-program av ett antal instruktioner. Till skillnad från BASIC-programmet så måste dock alla dessa instruktioner översättas till maskinkod, d v s den form av kod som datorns processor förstår, innan programmet kan köras. Denna översättning görs av ett särskilt program som kallas för en ASSEMBLERKOMPILATOR. Innan ett sådant program presenteras skall vi se lite på idéerna bakom programmet. Efter programlistningen följer sedan några exmpel på ASSEMBLER-program.

Om ASSEMBLER-kompilatorn är ett BASIC-program så vore det närliggande och möjligt att låta ASSEMBLER-programmet lagras i ett textfält (Initierat av t ex DIM A$ (999)). Ett sådant textfält kan sparas och hämtas från band eller disk. Den kompilator som presenteras här utnyttjar en annan lösning. Även ASSEMBLER-programmet lagras som ett BASIC-program! Det innebär att BASIC-editorns alla kommandon står till förfogande som LIST, SAVE, LOAD, instickning mellan raderna m m. Om man dessutom har programmers AID så finns MERGE, DELETE, RENUMBER, AUTO, automatisk scrollning av raderna upp och ned m m. Det innebär att kodningen blir både lätt och mångsidig. Om kodningen sker utan programmers AID så laddas först kompilatorn från band eller skiva. Därefter skrivs ASSEMBLER-programmet med första radnummer större än 767. Kompileringen sker därefter med kommandot RUN. Program (inklusive kompilator) kan sparas med SAVE. Exekvering sker efter kompilering med SYS.

FÖLJANDE KRAV

Den presenterade ASSEMBLER-kompilatorn uppfyller följande krav:

- Maskinkoden kan lokaliseras till godtycklig plats som ej utnyttjas av BASIC-programmet. (Ev. bör man skydda detta område genom att t ex ändra Top of Memory.)
- Långa variabelnamn.
- Upprepade additioner/subtraktioner möjliga i operanden.
- Relativa hoppadresser/variabler.
- Decimala eller HEXA-decimala tal och adresser får förekomma blandat.
- Indrag möjliga för att öka läsbarheten.
- Lagring av större dataareor i HEX-form.
- Lagring av ASC-värden av text.
- Förklarande texter.
- Listning av variabelnamn och hoppadressnamn med adresser.
- Adekvata felutskrifter.
- Möjlighet att skifta hög och låg byt.

För att möjliggöra detta finns följande extra instruktioner:

HEX

Möjliggör att i efterföljande bytar lagra valfria Hexa-decimala värden. T ex så lagrar instruktionen 1370 HEX A50063 de Hexa-decimala talen $A5, $00 och $63 i tre på varandra följande bytar. Detta är enda fallet då $-tecknet ej föregår Hexa-decimala tal. Operanden måste innehålla två tecken för varje HEX-värde.

ASC

Lagrar ASC-värdena av efterföljande text. T ex 1590 ASC ABC lagrar ASC-värdena av bokstäverna A, B och C d v s 65, 66 resp 67 i tre på varandra följande bytar.

REM

Används på samma sätt som i

BASIC för kommentarer. Kommentarer kan också skrivas efter instruktioner som endast upptar en byte maskinkod.

OBJ

Anger maskinkodens startadress. Denna instruktion bör förutom ev. REM- satser vara den första i programmet. OBJ kan också användas mitt i programmet och lokaliserar då resten av ASSEMBLER-programmets maskinkod till den nya startadressen. T ex
1020 OBJ $6000
sätter maskinkodens startadress till $6000 (HEXA-decimalt).

LBL

Label (etikett) för att definiera ett läge i programmet. T ex
1200 BCC HOPP 1
.
.
.
1350 LBL HOPP 1
På rad 1200 sker ett test om Carry är Clear (=0). Om så är fallet så hoppar exekveringen till raden under rad 1350. LBL används också för att definiera relativa variabler eller en datamängd.

LET

För att definiera adressen av en variabel (eller värdet av en konstant). Ingen variabel får användas utan att någonstans i programmet lokaliseras med en LET-sats eller en LBL-sats. LET-satser bör av åskådlighetsskäl placeras i början av programmet men detta är inget krav. T ex
1030 LET KVOT = 828
1040 LET ADRL = $87
medför att de två variablerna KVOT och ADRL får adresserna 828 (decimalt) respektive $87 (HEXA-decimalt).
Därefter är det möjligt att skriva en sats som
1050 LET ADRH=ADRL+3
vilken gör att variabeln ADRH får adressen $8A.

Till sist: ↑ används för att skifta hög och låg byt och sätts framför variabel/uttryck. T ex
2090 LDA #KVOT
laddar den låga byten av adressen för variabeln KVOT medan
2090 LDA # ↑KVOT
laddar den höga byten av adressen i ackumulatorn.

Kompileringen (som tar viss tid) sker i två steg. I första steget så beräknas alla adresser för variabler och hoppadresser och i det andra steget så skapas maskinkoden.

Vi skall nu avsluta med två exempel på ASSEMBLER-program. Syftet är inte att lära ut ASSEMBLER-programmering utan att se hur koden ser ut för vår speciella kompilator. För att lära sig ASSEMBLER-programmering se artikelserien ASSEMBLER skolan i denna tidskrift eller t ex den utmärkta boken Programming the 6502 av Rodnay Zaks. Skillnaden mot de monitorer som finns på marknaden är små. De största skillnaderna är att vi här har radnummer och att kommenteringen fungerar annorlunda. Med viss tacksamhet noterar man också att man slipper alla specialkommandon.

Båda programmen förutsätter 8K eller mer minnesexpansion men den intresserade programmeraren kan säkert modifiera programmen efter andra minneskonfigurationer. I bägge fall lokaliseras maskinkodens startadress till startadress 673. Det innebär att exekveringen sker med kommandot SYS 673. Ett bra sätt att spara den färdiga maskinkoden är att utnyttja programmet MINNESDATA i VIC rapport nr 4 1984.

Programmen är i stort sett självförklarande men lägg märke till att operanden kan dras in. Alla mellanslag räknas bort vid kompileringen vilket gör det lätt att strukturera programmet. Z-mode deklareras med ett Z framför operanden.

Åke Lundeqvist

Program "ASSEMBLERKOMPILATOR"

```
100 REM *** KOMPILERA AKE LUNDEQUIST 83-09-05 ***
102 PRINT"       KOMPILERA":GOSUB486
104 DEFFNH(X)=INT(X/U)ANDU4:DEFFNL(X)=X-U*INT(X/U)
106 PRINT"*** KOMPILERA FAS1 ***
108 P=PS:K=1024
110 GOSUB190:IFE>0THEN188
112 IFD<12THENK=K+L%(D):GOTO134
114 IFM$="END"THENV%(V)=K:V$(V)="END":GOTO136
116 IFM$="ASC"THENK=K+LEN(O$)
118 IFM$="HEX"THENK=K+LEN(O$)/2:GOTO134
120 IFM$="LBL"THENV$(V)=O$:V%(V)=K:V=V+1:GOTO134
122 IFM$="OBJ"THENGOSUB230:K=0:IFE>0THEN188
124 IFM$="WOR"THENK=K+2:GOTO134
126 IFM$<>"LET"THEN134
128 E=2:FORI=1TOLEN(O$):IFMID$(O$,I,1)="="THENX=I:I=99:E=0
130 NEXT:IFE>0THEN188
132 V$(V)=LEFT$(O$,X-1):O$=MID$(O$,X+1):GOSUB230:V%(V)=0:V=V+1:IFE>0THEN188
134 P=PN:GOTO110
136 PRINT"*** KOMPILERA FAS2 ***":P=PS:K=1024
138 GOSUB190:IFD<12THEN156
140 IFM$="END"THEN168
142 IFM$="WOR"THEN158
144 IFM$="ASC"THENFORI=1TOLEN(O$):POKEK,ASC(MID$(O$,I)):K=K+1:NEXT:GOTO166
146 IFM$="OBJ"THENGOSUB230:K=0:GOTO166
148 IFM$<>"HEX"THEN166
150 FORI=1TOLEN(O$)STEP2:C$=MID$(O$,I,2):GOSUB264
152 POKEK,FNL(C):K=K+1
154 NEXT:GOTO166
156 POKEK,ASC(MID$(K$(M),W,1)):K=K+1:IFL%(D)=1THEN166
158 GOSUB230:IFE>0THEN188
160 IFD=9THENO=O-1-K
162 POKEK,FNL(O):K=K+1:IFL%(D)=2THEN166
164 POKEK,FNH(O):K=K+1
166 P=PN:GOTO138
168 PRINT"KLART!":PRINT"VILL DU SE SYMBOL-    TABELLEN (J/N)?
170 GETA$:IFA$="N"THENEND
172 IFA$<>"J"THEN170
174 PRINT"SYMBOLTABELL:":PRINT"======================VARIABEL     DEC  HEX
176 FORI=0TOV:X=V%(I):A$=STR$(X):GOSUB268
178 PRINTV$(I)TAB(16-LEN(A$))A$"  "X$
180 GETA$:IFA$=""THEN180
```

Forts på sidan 10

Forts från sidan 5

>AGAIN
You aren't even holding the sword. The axe crashed against the rock, throwing sparks!

>LOOK
The Troll Room
This is a small room with passages to the east and south and a forbidding hole leading west. Bloodstains and deep scratches (perhaps made by an axe) mar the walls.
A nasty-looking troll, brandishing a bloody axe, blocks all passages out of the room.
The troll's axe barely misses your ear.

>KICK TROLL
Annoying the troll isn't notably helpful.
The troll swings; the blade turns on your armor but crashes broadside into your head.

>GO NORTH
You can't go that way.
The troll swings; the blade turns on your armor but crashes broadside into your head.

>GO WEST
The troll fends you off with a menacing gesture.
The troll swings his axe, but it misses.

>LOOK
The Troll Room



This is a small room with passages to the east and south and a forbidding hole leading west. Bloodstains and deep scratches (perhaps made by an axe) mar the walls.
A nasty-looking troll, brandishing a bloody axe, blocks all passages out of the room.
The troll's swing almost knocks you over as you barely parry in time.

>GO EAST
The troll fends you off with a menacing gesture.
The axe sweeps past as you jump aside.

>SCREAM
Aaaarrrrgggghhhh!
The troll's axe removes your head. It appears that that last blow was too much for you. I'm afraid you are dead.

You have died

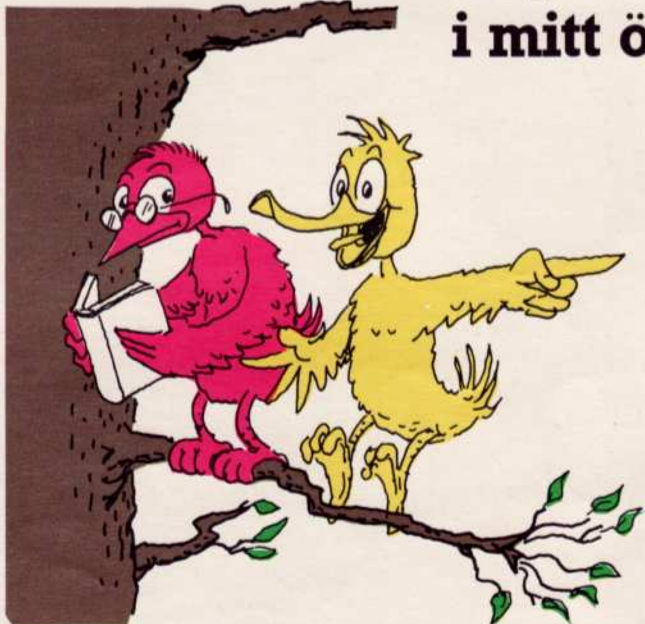
En fågel viskade i mitt öra...

Den stränga kylan har på ett paradoxalt sätt påverkat mina vänner Rykten och Sanningar.

Sanningar den trofasta och ärliga fågeln har påverkats av kylan så till vida att hon mest suttit hemma och drabbats av influensavågen. Därför har hon ingenting att rapportera om denna gång.

Rykten däremot har blommat upp under den stränga vinterkylan och är mer aktiv än någonsin. Som vanligt vill jag påpeka att allt som Rykten kvittrar om ska ifrågasättas. Det finns ingen som helst säkerhet för att det hon kvittrar om har någon förankring i verkligheten.

Rykten kvittrade även denna månad om att Commodore själva kommer att ta över verksamheten i Sverige. De kommer att styra denna verksamhet från Stockholm, närmare bestämt Spånga.



När försäljningen kommer att tas över av Commodore från Handic Electronic vågade inte ens rykten sia om. Handic Electronics roll i framtiden är också lite luddig. Rykten sa att de behåller programvaruförsäljningen medan Commodore håller i hårdvaran – men även Rykten garderade sig mot detta. Hur ett övertagande av VIC försäljningen kommer att påverka marknaden och oss slutanvändare är svårt att säga. Möjligen kan prisbilden påverkas gynnsamt i och med att ett distributionsled försvinner. Men framtiden är väl den som bäst kan avgöra om Rykten kvittrar sanning eller om det är kylan som påverkar hennes fantasi.

STOPIT

för VIC 64

Bra spel bygger som regel på en enkel idé. STOPIT är ett exempel på ett sådan spel som, trots sin enkelhet, kan få grepp om speldjävulen inom dig.

Spelet går ut på att få stopp på en grön liten boll som rör sig vertikalt och horisontellt på skärmen. Grundregeln är att man måste få stopp på bollen innan den når spelfältets ram. Bollen kan stoppas med vilken tangent som helst (mellanslagstangenten passar utmärkt). Lyckas man stoppa den innan den når ramen så ändrar den riktning.

Men detta är naturligtvis inte hela uppgiften. Det finns också ett antal siffror utplacerade på skärmen. Man samlar poäng genom att stoppa bollen mitt på siffrorna. De siffror som är mest värda (niorna) ligger naturligtvis farligt nära ramen...

En programmeringsteknisk "finess" med spelet är musiken som kan spela samtidigt som andra saker pågår. Detta är möjligt genom att använda en s k "interrupt". Hur programmet i övrigt är konstruerat kan man få en aning om genom att studera de REM-satser som finns instoppade här och var. □

G Berglund

```
10 REM ++++ STOPIT +++++
12 REM +++++++++++++++++
14 REM
40 GOSUB 7900:REM READ MUSIC
50 GOTO 3000: REM VINJETT
70 REM *** INITIERA ****
75 D(1)=-1:D(2)=1:D(3)=-40:D(4)=40:REM RIKTNINGAR
78 S=1524:CS=54272:D=50
80 RQ=INT(RND(1)*4)+1:Q=D(RQ):REM STARTRIKTNING
85 REM ***** END INIT *****
87 GOSUB 3150:REM RITA RAM
90 GOSUB 2000:REM PRINT NUMBERS & START GAME
95 LJ=30:POKECS+14,100:POKECS+20,240:POKECS+18,33:POKECS+24,15
98 SYS49901:REM MUSIC OFF
99 REM
100 REM ***** MAIN *****
101 REM
105 POKES,32:S=S+Q:PS=PEEK(S)
115 IFPS=32THENPOKES,81:POKES+CS,5
125 IFPS=64ORPS=93THEN400:REM GAME OVER
135 TT=PEEK(197):IFTT<>64THENGOSUB300
145 POKECS+15,LJ:LJ=LJ+3
165 FORT=0TOD:NEXT
175 GOTO105
195 REM **** END MAIN *****
196 REM
300 REM NY Q / SCORE?
310 RQ=INT(RND(1)*4)+1:LJ=30:D=50
320 IFQ=D(RQ)THEN310
325 LJ=30:POKECS+14,100:POKECS+20,240:POKECS+18,33:POKECS+24,15
330 Q=D(RQ)
340 REM HIT
345 IFPEEK(S)=81THEN390:REM MISS
348 PN=PEEK(S)-48:PG=PG+PN
350 POKECS+14,100:POKECS+19,15:POKECS+18,33:POKES,81
355 FORI=1TO3:POKES+CS,I
360 FORL=20TO40:POKECS+15,L:NEXT:NEXTI
370 FORL=20TO70:POKECS+15,L:NEXT:FORT=1TO50:NEXT
380 PRINTCU$"SCORE:"PG;
390 IFPEEK(197)<>64THEN390
399 RETURN
400 REM **** GAME OVER ***
410 POKES,32
420 REM LJUD
425 POKECS+14,100:POKECS+19,15:POKECS+18,129
430 FORI=0TO15:FORL=20TO40:POKECS+15,L:NEXT:POKECS+24,15-I:NEXT
435 FORL=14TO23:POKECS+L,0:NEXT:FORT=1TO500:NEXT
440 SYS49547:REM MUSIK
450 REM RESULTAT
460 GOSUB2210
470 PG=0:PRINTCU$"SCORE:"PG;
490 GOTO70:REM INITIERA
```

Programlistningen fortstätter på sidorna 44–45

Program ASSEMBLERKOMPILATOR

```
182 NEXT
184 PRINT"======================
186 END
188 PRINT"FEL "E$(E-1):PRINT"RAD NR"RN:END
190 PN=PEEK(P)+U*PEEK(P+1):RN=PEEK(P+2)+U*PEEK(P+3)
192 A$="":T%=-1:FORI=P+4TOPN-2
194 Q=PEEK(I):IFQ>127THENA$=A$+B$(Q-128):GOTO202
196 IFQ=34THENT%=NOT(T%):GOTO202
198 IFQ=32THENIFT%THEN202
200 A$=A$+CHR$(Q)
202 NEXT
204 M$=LEFT$(A$,3)
206 O$=MID$(A$,4)
208 IFM$="END"THEND=12:RETURN
210 M=32:Q=32
212 IFM$=M$(M)THEN218
214 Q=Q/2:IFQ<1THENE=1:RETURN
216 M=M+Q*((M$<M$(M))-(M$>M$(M))):GOTO212
218 D$=D$(P%(M)):W=LEN(D$)
220 IFO$=""ANDASC(D$)=0THENW=1
222 D=ASC(MID$(D$,W,1)):S=LEN(S$(D)):T=LEN(T$(D))
224 IFLEFT$(O$,S)<>S$(D)ORRIGHT$(O$,T)<>T$(D)THENW=W-1:GOTO222
226 O$=MID$(O$,1+S,LEN(O$)-S-T)
228 RETURN
230 O=0:IFO$=""THENRETURN
232 C$="":T=1
234 FORI=1TOLEN(O$)
236 A$=MID$(O$,I,1)
238 IFA$="+"THENGOSUB252:O=O+T*C:T=1:C$="":GOTO244
240 IFA$="-"THENGOSUB252:O=O+T*C:T=-1:C$="":GOTO244
242 C$=C$+A$
244 NEXT
246 GOSUB252:O=O+T*C:O=O+U1*(O>U2)
248 IFO>U2ORO<U3THENE=3
250 RETURN
252 IFC$>"↑"THENC$=MID$(C$,2):GOSUB252:C=U*FNL(C)+FNH(C):RETURN
254 IFC$<"A"THEN260
256 E=3:FORJ=0TOV:IFC$=V$(J)THENC=V%(J):E=0:J=V
258 NEXT:RETURN
260 IFC$>"/"THENC=VAL(C$):RETURN
262 C$=MID$(C$,2):IFC$=""THENE=3:RETURN
264 C=0:FORJ=1TOLEN(C$):X=ASC(MID$(C$,J,1))-48:X=X+7*(X>9):C=16*C+X:NEXT
266 RETURN
268 Y=X:X$=""
270 FORJ=0TO3
272 Q=YAND15:Q=Q-7*(Q>9)+48:X$=CHR$(Q)+X$:Y=INT(Y/16)
274 NEXT
276 RETURN
278 DATA???,00,0
280 DATAADC,6D65697D79617175,1
282 DATAAND,2D25293D39213135,1
284 DATAASC,07,0
286 DATAASL,0A0E061E16,2
288 DATABCC,90,3
290 DATABCS,B0,3
292 DATABEQ,F0,3
294 DATABIT,2C24,4
296 DATABMI,30,3
298 DATABNE,D0,3
300 DATABPL,10,3
302 DATABRK,00,5
304 DATABVC,50,3
306 DATABVS,70,3
308 DATACLC,18,5
310 DATACLD,D8,5
312 DATACLI,58,5
314 DATACLV,B8,5
316 DATACMP,CDC5C9DDD9C1D1D5,1
318 DATACPX,ECE4E0,6
320 DATACPY,CCC4C0,6
322 DATADEC,CEC6DED6,7
324 DATADEX,CA,5
326 DATADEY,88,5
328 DATAEOR,4D45495D59415155,1
330 DATAHEX,0F,0
332 DATAINC,EEE6FEF6,7
334 DATAINX,E8,5
336 DATAINY,C8,5
338 DATAJMP,4C6C,8
340 DATAJSR,20,9
342 DATALBL,13,0
344 DATALDA,ADA5A9BDB9A1B1B5,1
346 DATALDX,AEA6A2BEB6,10
348 DATALDY,ACA4A0BCB4,11
```

▶



```
350 DATALET,17,0
352 DATALSR,4A4E465E56,2
354 DATANOP,EA,5
356 DATAOBJ,1B,0
358 DATAORA,0D05091D19011115,1
360 DATAPHA,48,5
362 DATAPHP,08,5
364 DATAPLA,68,5
366 DATAPLP,28,5
368 DATAREM,1F,0
370 DATAROL,2A2E263E36,2
372 DATAROR,6A6E667E76,2
374 DATARTI,40,5
376 DATARTS,60,5
378 DATASBC,EDE5E9FDF9E1F1F5,1
380 DATASEC,38,5
382 DATASED,F8,5
384 DATASEI,78,5
386 DATASTA,8D859D998191 95,12
388 DATASTX,8E8696,13
390 DATASTY,8C8494,14
392 DATATAX,AA,5
394 DATATAY,A8,5
396 DATATSX,BA,5
398 DATATXA,8A,5
400 DATATXS,9A,5
402 DATATYA,98,5
404 DATAWOR,FF,0
406 DATAL
408 DATAABCDEFGH
410 DATA@ABDH
412 DATAI
414 DATAAB
416 DATA@
418 DATAABC
420 DATAABDH
422 DATAAJ
424 DATAA
426 DATAABCEK
428 DATAABCDK
430 DATAABDEFGH
432 DATAABK
434 DATAABH
436 DATA ,,1
438 DATA ,,3
440 DATA Z,,2
442 DATA #,,2
444 DATA ,",X",3
446 DATA ,",Y",3
448 DATA (,",X)",2
450 DATA (,"),Y",2
452 DATA Z,",X",2
454 DATA ,,2
456 DATA (,),3
458 DATA Z,",Y",2
460 DATA ,,0
462 DATA MNM
464 DATA DEF
466 DATA TAL
468 DATA MOD
470 DATA HOPP
472 DATA TOK
474 DATA END,FOR,NEXT,DATA,INPUT#,INPUT,DIM,READ,LET,GOTO,RUN,IF
476 DATA RESTORE,GOSUB,RETURN,REM,STOP,ON,WAIT,LOAD,SAVE,VERIFY
478 DATA DEF,POKE,PRINT#,PRINT,CONT,LIST,CLR,CMD,SYS,OPEN,CLOSE
480 DATA GET,NEW,TAB(,TO,FN,SPC(,THEN,NOT,STEP,+,-,*,/,↑,AND,OR
482 DATA >,=,<,SGN,INT,ABS,USR,FRE,POS,SQR,RND,LOG,EXP,COS,SIN
484 DATA TAN,ATN,PEEK,LEN,STR$,VAL,ASC,CHR$,LEFT$,RIGHT$,MID$
486 DIMV$(255),V%(255),M$(63),K$(63),P%(63),D$(14)
488 DIM S$(12),T$(12),L%(12),B$(74),E$(5)
490 FORI=0TO63:READM$(I),A$,P%(I)
492 FORJ=1TOLEN(A$)STEP2
494 X=ASC(MID$(A$,J,1))-48:Y=ASC(MID$(A$,J+1,1))-48:X=X+7*(X>9):Y=Y+7*(Y>9)
496 K$(I)=K$(I)+CHR$(16*X+Y)
498 NEXTJ,I
500 FORI=0TO14:READA$:FORJ=1TOLEN(A$):X=ASC(MID$(A$,J,1))-64:D$(I)=D$(I)+CHR$(X)
502 NEXTJ,I
504 FORI=0TO12:READS$(I),T$(I),L%(I):NEXT
506 FORI=0TO5:READE$(I):NEXT
508 FORI=0TO74:READB$(I):NEXT
510 U=256:U1=65536:U2=32767:U3=-32768:U4=255:PS=43
512 PS=PEEK(PS)+U*PEEK(PS+1):RN=PEEK(PS+3):IFRN<3GOTO512
514 RETURN
READY.
```

□

Med subrutinen låter VIC 64 riktigt bra

Förra gången talade vi om SID-chipet i VIC 64 och gav ett program för framtagning av korrekta frekvenstal för en svensk VIC 64. Nu skall vi se litet på hur man kan skriva egna procedurer (subrutiner) och använda de så framtagna värdena för att få VIC 64an att låta riktigt bra. Dessutom skall vi se lite på hur man kan utnyttja procedurer för att göra vissa standardinställningar av de annars så komplicerade rutinerna som styr SID-chipet.

LÄR DIG PROGRAMMERA



COMAL-KURSEN fortsätter

SID-minnet

Adress 54272 — STÄMMA 1

F	R	E	K	V	E	N	S
R	E	G	I	S	T	E	R
			PULSBRÄDD				
–	–	–	–	PULSBRÄDD			
BRUS	⊓⊔⊓	/|/|	/\/\	TEST	RINGM	SYNC	GRIND
ATTACK				DECAY			
SUSTAIN				RELEASE			

Adress 54272+7 — STÄMMA 2

F	R	E	K	V	E	N	S
R	E	G	I	S	T	E	R
			PULSBRÄDD				
–	–	–	–	PULSBRÄDD			
BRUS	⊓⊔⊓	/|/|	/\/\	TEST	RINGM	SYNC	GRIND

Adress 54272+14 — STÄMMA 3

F	R	E	K	V	E	N	S
R	E	G	I	S	T	E	R
			PULSBRÄDD				
				PULSBRÄDD			
BRUS	⊓⊔⊓	/|/|	/\/\	TEST	RINGM	SYNC	GRIND

				VOLYMEN			

$1=2^7=128$	$1=2^6=64$	$1=2^5=32$	$1=2^4=16$	$1=2^3=8$	$1=2^2=4$	$1=2^1=2$	$1=2^0=1$
$1=2^7*256$	$1=2^6*256$	$1=2^5*256$	$1=2^4*256$	$1=2^3*256$	$1=2^2*256$	$1=2^1*256$	$1=2^0*256$

Vi börjar med att titta på SID-chipet igen. Som framgår av bilden har den gamla illustrationen kompletterats med ytterligare detaljer. Några kvarstår som tomma fält, men dem tar vi upp i en senare artikel och visar dem därför inte nu.

Det nya är att vi lagt till de två register, där man bestämmer det tonstyrande komplexa begrepp som på engelska kallas ENVELOPE. På svenska skulle det bli OMSLAG och det är precis vad det handlar om. Man talar om ATTACK och menar då själva anslagsförloppet, dvs hur tonen startar och stiger till sin högsta nivå (kallas också STIGTID). Man talar om DECAY och menar då hur tonen liksom "löses upp, minskas" ner till en viss nivå, som på engelska heter SUSTAIN och på svenska HÅLLNIVÅ. Detta är den enda faktorn i de fyra komponenterna av ENVELOPE som inte är tidsförlopp utan en nivå, ett bestämt höjdläge. Till slut skall tonen sluta att ljuda och dess avveckling kallas på engelska för RELEASE, på svenska talar man om AVKLINGNING.

En av de stora fördelarna med SID-chipet är att man kan styra samtliga dessa fyra faktorer för var och en av de tre stämmorna. Som synes av illustratonen är det minnesregistren nr (54272) +5 och +6 för stämma 1, nr +12 och +13 för stämma 2 och nr +19 och +20 för stämma 3, där dessa värden läggs in. Själva konstruktionen är ganska intressant, man har tagit och delat 8-bitars-orden i två s k nybbles eller halvord och låter dem fungera som 4-bitars ord. Det betyder, att varje 4-bitars nybble max kan inrymma värden upp till 15, jämför med det förstorade dubbelordsminnet i bilden ovan, där det visas vilka värden som läggs in med en binär 1:a i de olika bitarna. Övre radens fyra bitar från höger visar, att värdet på en binär 1:a i första biten är "1", en binär 1:a i andra biten är värd "2" och i tredje biten "4". I fjärde biten har den binära 1:an värdet "8". Om man alltså "sätter" alla fyra bitarna får halvordet värdet 8+4+2+1=15. Och 15 är just maximalt värde för faktorerna ATTACK, DECAY, SUBSTAIN och RELEASE.

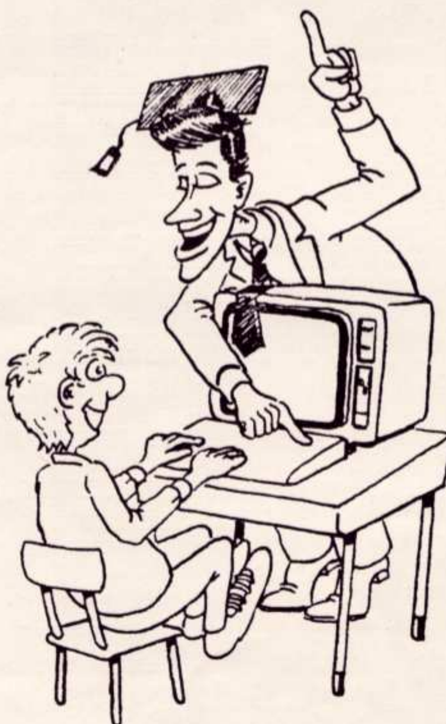
REGISTER FÖR PULSBREDD

De andra registren som vi skall beskriva här är registren för pulsbredd, dvs register nr (54272) +2 och +3 för stämma 1, +9 och +10 för stämma 2 och +16 och +17 för stämma 3. Här används hela 8-bitarsordet i registren +2, +9 och +16 för den "låga" delen av talet för pulsbredd. Den "höga" delen av talet som krävs för att ge pulsbredden är däremot bara ett halvt ord, en nybble och består alltså av fyra bitar. Däri kan alltså läggas en maximalvärde av
8*256+4*256+2*256+256+255 eller 4095. Pulsbredden är endast av intresse när man väljer fyrkantvåg, dvs "sätter" motsvarande bit i registren +4, +11 och +18 (fortfarande alltså utgående från SID-chipets start i minnesposition 54272).

Genom att sätta pulsbredden till halva maximivärdet får vi en fyrkantvåg med ungefär samma förlopp som en vanlig sinuskurva fast kantig. Detta är givetvis endast ett sätt att beskriva vågformen, i verkligheten ser den säkerligen ut som en sinusvåg. Genom att ändra pulsbredden kan man få ändrad klangfärg. Man bör dock observera att pulsbredd är en symmetrisk företeelse, vilket betyder att 0 och 4095 i princip ger samma kurva, dvs låter likadant. ▶

Slut på lagret

VIC rapport årg 3 nr 8 och 9 samt nr 1/85 är slut på lagret. Går ej att beställa.



SLUT PÅ MUSIK-TEORIN. Dags att programmera I COMAL-PROCEDURER

I COMAL finns en kraftfull konstruktion som kallas PROC eller procedurer. Det är del- eller subrutiner, som utför en bestämd uppgift och som anropas med namn i ett huvudprogram. Samtidigt med anropet kan man i COMAL också föra över parametervärden från sitt huvudprogram. Detta är en stor fördel eftersom procedurer i COMAL kan vara slutna, CLOSED. En CLOSED procedur kan använda samma variabelnamn som huvudprogrammet – utan att störa eller påverka dem i huvudprogrammet. Detta är ju inte möjligt i BASIC! Genom att introducera värden på i proceduren använda variabler och konstanter via anropssatsen, kan man också få tillbaka ändringar i dessa värden, som orsakas av bearbetningen inom proceduren.

Vi skall inte prata mer om detta utan går rätt på att visa, hur en OPEN, dvs inte CLOSED procedur kan användas för att lägga in de korrekta tonvärden, som vi räknade fram i förra artikelns program. Den var införd i nr 10/1984 av VIC rapport.

Låt oss göra proceduren så, att vi kan använda den i många olika musikprogram för att lägga in värdena som ger oss en rätt stämd VIC 64-musikmaskin. Vi kallar proceduren för LJUDINITIERING och skriver först det s k ”procedurhuvudet”.

PROC LJUDINITIERING (VOLYM)

Parentesen med variabeln ”volym” lades till för att enkelt låta oss bestämma volymen, dvs hur starkt tonerna skall höras. Visserligen kommer vi här in på en del av SID-chipet som vi inte berört i texten om SID ovan, men vi fördjupar oss i detta i en senare artikel. För ögonblicket gäller endast att kunna höra stämmorna och vi har en minnesposition där man kan lägga in maximum talet 15 med binära 1:or i fyra bitar. Det är det lägst liggande halvordet i position SID+24, där man lägger in volymvärdet.

Efter procedurhuvudet skall vi ge programmet möjlighet att hitta SID-chipet och det sker med tilldelningssatsen

```
SID:=54272 //Erfarna programmerare brukar i
             stället skriva SID:=13,25*4096,
             men det är samma sak!
```

För att hålla reda på våra notnamn måste vi också ha en strängmatris med 3 lägen för var och en av 96 toner, som är det maximala antalet toner som kan återges med SID. Det sker i en DIMensioneringssats, där vi också lägger in en DIM-sats för en numerisk vektor med lika många värden men med plats för två tal:

```
DIM TON$ OF 3*96, VÄRDEN(0:95,2)
```

Nu kan det vara lämpligt att bestämma pulsbredd och omslagskurva (ENVELOPE) för de tre stämmorna. Vi gör detta i enradiga FOR-slingor (sådana som inte behöver avslutas med ENDFOR eller NEXT därför att de endast är en rad långa). Genom att välja startvärde och stegvärde får vi bekvämt fram rätta minnespositionerna:

```
FOR I:=2 TO 16 STEP 7 DO POKE SID+I,0
FOR I:=3 TO 17 STEP 7 DO POKE SID+I,8
FOR I:=5 TO 19 STEP 7 DO POKE SID+I,6*16+12
FOR I:=6 TO 20 STEP 7 DO POKE SID+I,0
```

Skrivningen på raden med tredje FOR-slingan kanske skall förklaras litet. Genom att använda detta sätt att skriva, underlättar man läsningen av det värde man gett in till ATTACK och DECAY. Vi minns från det tidigare att vi arbetar med halvord, dvs 4-bitars s k nybbles. Det mest signifikanta halvordets bitar utgör egentligen multiplar av 16, jämför SID-bildens förstorade del. Genom att då skriva 6*16+12 framgår tydligt at vi lägger in värdet 6 i det högre halvordet och i det lägre värdet 12. Naturligtvis får vi samma resultat om vi räknar ut 6*16+12 och lägger in 108 decimalt till helordet SID+5! Fördelen med ovanstående skrivsätt är att man direkt ser både ATTACK- och DECAY-värdena.

STÄNGA MED GRINDEN

När vi nu skall börja lägga in tonvärden vill vi väl inte att SID skall ”låta” under tiden detta sker. Därför stänger vi av ljudutgången genom att sätta den s k ”grinden” eller GATE till noll för varje stämma. Det sker i en extra FOR-snurra:

```
FOR I:=4 TO 18 STEP 7 DO POKE SID+I,0
```

Och äntligen är vi framme vid att läsa in värdena från de DATA-satser som vi fått fram ur det tidigare programmet.

Naturligtvis skulle vi ha kunnat bygga på det gamla programmet så att dess värden läses direkt in till SID, men det är ju en kombination som läsaren själv kan arbeta med, eller hur?

Läsningen skall ju upprepas ett bestämt antal gånger dvs 96 eller lika många som det finns tonvärden. Det gör att en FOR-slinga passar bra. Vi skriver alltså:

```
FOR I:=0 TO 95 DO
```

för att inleda slingan, stegvärdet är här automatiskt 1 när inget annat värde anges.

```
READ TON$(I*3+1:I*3+3),VÄRDEN(I,1),VÄRDEN(I,2)
```

Själva avläsningen görs med en READ-sats där vi använder I-värdet för att få rätt plats i DATA-satserna:

Fortsättning på sidan 22

Program "COMAL-SKOLAN"

OBS! ¤ = $

```
ØØ1Ø // LIST "LJUDINIT.L"
ØØ2Ø PROC LJUDINITIERING(VOLYM)
ØØ3Ø  SID:=13.25*4Ø96
ØØ4Ø  DIM TON¤ OF 3*96, VÄRDEN(Ø:95,2)
ØØ5Ø  //
ØØ6Ø  POKE SID+24,VOLYM
ØØ7Ø  //
ØØ8Ø  FOR I:=2 TO 16 STEP 7 DO POKE SID+I,Ø
ØØ9Ø  FOR I:=3 TO 17 STEP 7 DO POKE SID+I,8
Ø1ØØ  FOR I:=5 TO 19 STEP 7 DO POKE SID+I,6*16+12
Ø11Ø  FOR I:=6 TO 2Ø STEP 7 DO POKE SID+I,Ø
Ø12Ø  FOR I:=4 TO 18 STEP 7 DO POKE SID+I,Ø
Ø13Ø  //
Ø14Ø  FOR I:=Ø TO 95 DO
Ø15Ø   READ TON¤(I*3+1:I*3+3),VÄRDEN(I,1),VÄRDEN(I,2)
Ø16Ø  ENDFOR I
Ø17Ø  //
 18Ø  DATA "CØ ",1,13
Ø19Ø  DATA "CØ#",1,3Ø
Ø2ØØ  DATA "DØ ",1,48
Ø21Ø  DATA "DØ#",1,67
Ø22Ø  DATA "EØ ",1,87
 23Ø  DATA "FØ ",1,1Ø8
Ø24Ø  DATA "FØ#",1,13Ø
Ø25Ø  DATA "GØ ",1,153
Ø26Ø  DATA "GØ#",1,178
Ø27Ø  DATA "AØ ",1,2Ø4
Ø28Ø  DATA "AØ#",1,232
Ø29Ø  DATA "HØ ",2,6
Ø3ØØ  DATA "C1 ",2,37
Ø31Ø  DATA "C1#",2,7Ø
Ø32Ø  DATA "D1 ",2,1Ø5
Ø33Ø  DATA "D1#",2,142
Ø34Ø  DATA "E1 ",2,181
Ø35Ø  DATA "F1 ",2,223
Ø36Ø  DATA "F1#",3,12
Ø37Ø  DATA "G1 ",3,59
Ø38Ø  DATA "G1#",3,1Ø9
Ø39Ø  DATA "A1 ",3,162
 4ØØ  DATA "A1#",3,218
Ø41Ø  DATA "H1 ",4,22
Ø42Ø  DATA "C2 ",4,85
Ø43Ø  DATA "C2#",4,151
Ø44Ø  DATA "D2 ",4,221
Ø45Ø  DATA "D2#",5,4Ø
Ø46Ø  DATA "E2 ",5,11Ø
Ø47Ø  DATA "F2 ",5,2Ø3
Ø48Ø  DATA "F2#",6,37
Ø49Ø  DATA "G2 ",6,131
Ø5ØØ  DATA "G2#",6,23Ø
Ø51Ø  DATA "A2 ",7,8Ø
Ø52Ø  DATA "A2#",7,191
Ø53Ø  DATA "H2 ",8,54
Ø54Ø  DATA "C3 ",8,179
Ø55Ø  DATA "C3#",9,57
Ø56Ø  DATA "D3 ",9,197
Ø57Ø  DATA "D3#",1Ø,91
Ø58Ø  DATA "E3 ",1Ø,249
Ø59Ø  DATA "F3 ",11,161
Ø6ØØ  DATA "F3#",12,83
Ø61Ø  DATA "G3 ",13,16
Ø62Ø  DATA "G3#",13,215
Ø63Ø  DATA "A3 ",14,171
Ø64Ø  DATA "A3#",15,139
Ø65Ø  DATA "H3 ",16,12Ø
Ø66Ø  DATA "C4 ",17,115
Ø67Ø  DATA "C4#",18,125
Ø68Ø  DATA "D4 ",19,151
Ø69Ø  DATA "D4#",2Ø,194
Ø7ØØ  DATA "E4 ",22,Ø
Ø71Ø  DATA "F4 ",23,79
Ø72Ø  DATA "F4#",24,178
Ø73Ø  DATA "G4 ",26,43
Ø74Ø  DATA "G4#",27,186
Ø75Ø  DATA "A4 ",29,97
Ø76Ø  DATA "A4#",31,32
Ø77Ø  DATA "H4 ",32,249
Ø78Ø  DATA "C5 ",34,239
Ø79Ø  DATA "C5#",37,4
Ø8ØØ  DATA "D5 ",39,55
Ø81Ø  DATA "D5#",41,14Ø
Ø82Ø  DATA "E5 ",44,5
Ø83Ø  DATA "F5 ",46,162
Ø84Ø  DATA "F5#",49,1Ø4
Ø85Ø  DATA "G5 ",52,88
Ø86Ø  DATA "G5#",55,117
Ø87Ø  DATA "A5 ",58,193
 88Ø  DATA "A5#",62,64
Ø89Ø  DATA "H5 ",65,243
Ø9ØØ  DATA "C6 ",69,223
Ø91Ø  DATA "C6#",74,8
Ø92Ø  DATA "D6 ",78,111
 93Ø  DATA "D6#",83,26
Ø94Ø  DATA "E6 ",88,11
Ø95Ø  DATA "F6 ",93,71
Ø96Ø  DATA "F6#",98,211
Ø97Ø  DATA "G6 ",1Ø4,18Ø
Ø98Ø  DATA "G6#",11Ø,238
Ø99Ø  DATA "A6 ",117,136
1ØØØ  DATA "A6#",124,134
1Ø1Ø  DATA "H6 ",131,238
1Ø2Ø  DATA "C7 ",139,199
1Ø3Ø  DATA "C7#",148,24
1Ø4Ø  DATA "D7 ",156,23Ø
1Ø5Ø  DATA "D7#",166,6Ø
1Ø6Ø  DATA "E7 ",176,32
1Ø7Ø  DATA "F7 ",186,154
1Ø8Ø  DATA "F7#",197,18Ø
1Ø9Ø  DATA "G7 ",2Ø9,119
11ØØ  DATA "G7#",221,237
111Ø  DATA "A7 ",235,34
112Ø  DATA "B7 ",249,31
113Ø  DATA "Z  ",Ø,Ø
114Ø  //
```

Listningen fortsätter på sidan 16

Forts "COMAL-SKOLAN"

OBS! ¤ = $

```
115Ø ENDPROC LJUDINITIERING
116Ø // LIST "OMSLAG.L"
117Ø PROC OMSLAG(STÄMMA,STIGTID,SJUNKNING,HÅLLNIVÅ,AVKLINGNING)
118Ø  J:=SID+(STÄMMA-1)*7
119Ø  POKE J+5,STIGTID*16+SJUNKNING
12ØØ  POKE J+6,HÅLLNIVÅ*16+AVKLINGNING
121Ø ENDPROC OMSLAG
122Ø // LIST "GUBBEN'NOAK.L"
123Ø //
124Ø // PROGRAM EFTER IDEER FRÅN UNICOMAL APS, DANMARK
125Ø // ÄNDRAT AV ÅKE FREDRIKSSON, COMALKLUBBEN I SVERIGE
126Ø //
127Ø LJUDINITIERING(15)
128Ø DIM N¤ OF 3
129Ø S:=SID
13ØØ FOR II:=1 TO 3 DO
131Ø  OMSLAG(II,Ø,4,12,1Ø)
132Ø ENDFOR II
133Ø KONTROLL:=32
134Ø REPEAT
135Ø  N¤:="   "
136Ø  READ N¤(1:3),LNG
137Ø  IF N¤<>"XX " THEN
138Ø   IF N¤<>"Z  " THEN
139Ø    N¤(2):=CHR¤(ORD(N¤(2))+4)
14ØØ    I:=((N¤ IN TON¤)-1)/3
141Ø    POKE S,VÄRDEN(I,2)
142Ø    POKE S+1,VÄRDEN(I,1)
143Ø    POKE S+4,KONTROLL+1
144Ø   ELSE
145Ø    POKE S+4,KONTROLL
146Ø    FOR II:=1 TO 4Ø DO NULL
147Ø   ENDIF
148Ø   FOR II:=1 TO LNG*4Ø DO NULL
149Ø   POKE S+4,KONTROLL
15ØØ   S:+7
151Ø   IF S=SID+3*7 THEN S:=SID
152Ø  ENDIF
153Ø UNTIL EOD
154Ø POKE SID+24,Ø
155Ø END
156Ø //
157Ø DATA "F1",8,"Z",2,"F1",8,"Z",2,"F1",8,"Z",2,"A1",8,"Z",2
158Ø DATA "G1",8,"Z",2,"G1",8,"Z",2,"G1",8,"Z",2,"A1#",8,"Z",2
159Ø DATA "A1",8,"Z",2,"F1",8,"Z",2,"G1",8,"Z",2,"E1",8,"Z",2
16ØØ DATA "F1",2Ø,"Z",24
161Ø DATA "A1",8,"Z",2,"A1",8,"Z",2,"A1",8,"Z",2,"C2",8,"Z",2
162Ø DATA "C2",1Ø,"A1#",1Ø,"Z",2,"A1#",8,"Z",2,"A1#",8,"Z",2
163Ø DATA "G1",8,"Z",2,"G1",8,"Z",2,"G1",8,"Z",2,"A1#",8,"Z",2
164Ø DATA "A1#",1Ø,"A1",1Ø,"Z",2,"A1",2Ø,"Z",2
165Ø DATA "F1",8,"Z",2,"F1",8,"Z",2,"F1",8,"Z",2,"A1",8,"Z",2
166Ø DATA "G1",8,"Z",2,"G1",8,"Z",2,"G1",8,"Z",2,"A1#",8,"Z",2
167Ø DATA "A1",8,"Z",2,"F1",8,"Z",2,"G1",8,"Z",2,"E1",8,"Z",2
168Ø DATA "F1",4Ø,"XX",Ø
```

□

Sprite Editor

För alla er som har en VIC 64 kan det vara till glädje att ha en Sprite Editor. Genom denna spar du både papper och penna.

Programmet i sig är ganska långt och det är därför viktigt att vara noga med "inknappandet". Jäkta inte för då kan det lätt uppstå något fel. Programmet kvaddas om någon printsats är fel. Var gärna lite extra noga med datasatserna 2010 och 2020.

TEST AV SPRITE EDITOR

Starta upp programmet med RUN. Då får du upp huvudmenyn där du trycker på tangent 1. Nu visas själva sprite Editorn i funktion. I det nedre hörnet ser du en fyrkant, dvs en sprite. Där kommer även dina egna sprites att hamna. Prova nu med att göra en egen figur i matrisen, du använder dig då av tangenterna S, mellanslag och cursortangenterna. Om du skulle vara osäker på tangenterna så tittar du bara i kommandolistan.

När du väl är klar med din figur så trycker du på tangenten = och då ska din figur synas i rutan (om inte så har du varit slarvig när du knappat in programmet). Tryck på tangent C och din figur försvinner.

Om du nu trycker på F1 så ser du i matrisen att din sprites byggs upp igen.

När detta är klart så tryck på M (som i meny) och här trycker du på tangent 4. Datorn frågar nu om du har ångrat dig – du tar fram ett band och återspolar det – när detta är klart trycker du på N (som i nej) och därefter trycker du på RECORD/PLAY på bandspelaren. Efter någon minut är allt klart.

När du stoppar programmet använder du dig av RUN STOP och RESTORE tangenterna. När du sedan vill starta programmet igen upptäcker du att din sprite inte existerar längre. Tryck då på tangent 3 i menyn och ännu en gång frågar datorn om du ångrat dig – spola då tillbaka bandet – tryck sedan på tangenten N – och sedan på PLAY på bandspelaren. När detta är klart trycker du på tangent 1 i menyn och du finner då din sprite i det nedre hörnet. Tryck på F1 och du finner din figur i matrisen precis som du lämnade den. □

Joakim Krassman

(Kommando tangent)

RESUL.(=)

Du får se hur din figur blev. Det tar en liten stund innan du får se spriten. Cursorn försvinner för att markera att Datorn arbetar på ett annat håll. När allt är klart poppar spriten ut med din figur.

X LED:Y LED.(X),(Y)

Expanderar spriten i X och i Y ledet. Eller om du vill, på höjden och bredden.

MENY.(M)

Återgår till huvudmenyn.

SUDDA.(S)

Suddar ut den befintliga rutan som cursorn står på.

EX,FÄRG 1:EX,FÄRG 2.(←),(↑)

Lägger på extra färg på spriten. När du valt 15 färger så börjar de om från början med svart.

COL.(@)

Är huvudfärgen på spriten. Om du använder extra färger så kan du max få tre färger i taget.

CLR.(C)

Suddar ut hela matrisen, dvs den du gör ditt mönster på.

DATA.(D)

Samtliga datasatser kan studeras om och om igen. Du kan även få dem utprintade.

HOME.(H)

Flyttar upp cursorn i det vänstra hörent.

F1.(F1)

Kopierar ut den synliga spriten i rutan över till den stora matrisen.

F3.(F3)

ON/OFF läge för extra färgerna.

(Mellan slag)

Plottar ut en punkt där cursorn för tillfället befinner sig.

(Cursor tangenterna)

Flyttar cursorn.

Programmet finns på sidorna 18, 19 och 23

Program "SPRITE EDITOR"



```
READY.

1 REM************************************
2 REM***     HEAVY SPRITE EDITOR.     ***
3 REM***            AV                ***
4 REM***      JOAKIM  KRASSMAN        ***
5 REM***   (C)-84 COPY RIGHT 1984     ***
6 REM************************************
10 PRINT"▒▒▒":POKE53280,0:POKE53281,0:GOSUB1140:GOSUB2000
15 DIMF(64),FF(64),SA(64):GOSUB4000:GOTO4040
20 GOSUB1000:GOSUB600:X=33:Y=60:P=1105:J=P:FF=1:POKEV,X:POKEV+1,Y:POKEV+21,255
97 REM************************************
98 REM***      HUVUD PROGRAM.         ***
99 REM************************************
100 GETA$:IFA$=""THEN100
110 IFA$="Y"THENGOSUB800:REMPOKEV+2,245:POKEV+3,185:POKEV+23,2:POKEV+29,2
115 IFA$="X"THENGOSUB820:REM POKEV+2,255:POKEV+3,195:POKEV+23,0:POKEV+29,0
120 IFA$="▒"THENX=X+8:X=X:P=P+1:GOTO180
125 IFA$="←"THENM=PEEK(53285)AND15:M=M+1:GOSUB500
130 IFA$="┐"THENY=Y-8:Y=Y:P=P-40:GOTO210
133 IFA$="M"THEN4040
135 IFA$="↑"THENA1=PEEK(53286)AND15:A1=A1+1:GOSUB520
140 IFA$="▒"THENY=Y+8:Y=Y:P=P+40:GOTO200
142 IFA$="H"THENX=33:Y=60:P=1105:J=P:POKEV,X:POKEV+1,Y
145 IFA$="D"THEN3000
150 IFA$="▌"THENX=X-8:X=X:P=P-1:GOTO190
155 IFA$="@"THENFF=FF+1:POKEV+40,FF:IFFF>15THENFF=1:POKEV+40,FF
160 IFA$=" "THENGOSUB240
165 IFA$="■"THENWE=PEEK(53276):GOSUB550
166 IFA$="C"THENGOSUB760
170 IFA$="="THENU=1:POKEV,0:POKEV+1,0:GOTO250
173 IFA$="■"THENC=0:S=0:L=0:GOTO630
175 IFA$="S"THENPOKEP,79:POKEP+54272,2
176 GOTO100
180 IFX>217THENX=X-8:Y=Y:P=P-1:GETA$:GOTO120
190 IFX<33THENX=X+8:Y=Y:P=P+1:GETA$:GOTO150
200 IFY>220THENY=Y-8:X=X:P=P-40:GETA$:GOTO140
210 IFY<60THENY=Y+8:X=X:P=P+40:GETA$:GOTO130
220 POKEV,X:POKEV+1,Y
230 GOTO100
240 POKEP,160:POKEP+54272,2:RETURN
241 REM**********************************
242 REM** BERAKNING AV DATASATSERNA **
243 REM**********************************
250 B=128:Z=1:FORI=1TO64:F(I)=0:NEXT:Q=X:W=Y
260 FORH=0TO23
270 IFPEEK(W(Z)+H)=160THENF(U)=F(U)+B
280 IFB=1THENB=128:U=U+1:GOTO300
290 B=B/2
300 NEXTH
310 Z=Z+1:IFZ>=22THENC=0:GOTO340
320 B=128:GOTO260
340 POKE2041,201:FORD=64*201TO64*201+62:C=C+1:POKED,F(C):NEXT
350 C=0:POKEV,X:POKEV+1,Y:GOTO100
497 REM**********************************
498 REM**RUTINER AV EXTRA VALD FARG.**
499 REM**********************************
500 IFM>15THENM=1
510 POKE53285,MOR240:RETURN
520 IFA1>15THENA1=1
530 POKE53286,A1OR240:RETURN
550 IFWE<>0THEN580
560 POKE53276,2
570 RETURN
580 POKE53276,0:RETURN
597 REM**********************************
598 REM***     POKAR UT PR SPRITEN    ***
599 REM**********************************
600 POKEV+2,245:POKEV+3,185:POKEV+23,2:POKEV+29,2:POKE2041,201:POKEV+21,3
610 POKEV+40,1:RETURN
620 REM**********************************
621 REM**KOPIERAR SPRITEN I MATRISEN.**
622 REM**********************************
630 FORS=0TO63:FF(S)=F(S):NEXTS
640 L=L+1:S=0
650 FORH=1TO3
660 C=C+1
670 IFFF(C)<>0THENB=128:GOTO690
680 S=S+8:NEXTH:GOTO730
690 FORT=0TO7
700 IFFF(C)>=BTHENPOKEW(L)+S,160:POKEW(L)+S+54272,2:FF(C)=FF(C)-B
710 S=S+1:B=B/2
720 NEXT:NEXT
730 IFL<21THEN640
740 GOTO100
750 REM**********************************
751 REM**      RENSA MATRISEN.        **
752 REM**********************************
760 PRINT"▒▒▒"
770 FORI=1TO21:PRINT"▒▒┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬ ":NEXT
780 PRINT"▒────────────────────────"
790 RETURN
791 REM**********************************
792 REM***     EXPANDERA I Y LED.     ***
793 REM**********************************
800 IFPEEK(V+23)<>0THENPOKEV+23,0:POKEV+3,185:RETURN
```

▶

```
810 POKEV+23,2:POKEV+2,245:RETURN
811 REM*********************************
812 REM***    EXPANDERA I X LED.     ***
813 REM*********************************
820 IFPEEK(V+29)<>0THENPOKEV+29,0:POKEV+3,185:RETURN
830 POKEV+29,2:POKEV+3,185:RETURN
997 REM*********************************
998 REM**    BILDSKARMS RUTINEN.      **
999 REM*********************************
1000 FORI=1TO21:PRINT"■TTTTTTTTTTTTTTTTTTTTTTTTT":NEXT
1010 PRINT"■───────────────────────"
1020 PRINT"■HEAVY SPRITE EDITOR AV JOAKIM KRASSMAN. "
1030 PRINTTAB(26)"■=■ RESUL
1040 PRINTTAB(26)"■X■ X LED
1045 PRINTTAB(26)"■Y■ Y LED
1050 PRINTTAB(26)"■M■ MENY"
1055 PRINTTAB(26)"■S■ SUDDA
1060 PRINTTAB(26)"■←■ EX.FARG 1
1070 PRINTTAB(26)"■↑■ EX.FARG 2
1080 PRINTTAB(26)"■■ COL
1090 PRINTTAB(26)"■C■ CLR
1095 PRINTTAB(26)"■D■ DATA
1096 PRINTTAB(26)"■H■ HOME"
1097 PRINTTAB(26)"■F1■ M COPY
1100 PRINTTAB(26)"■F3■ EXTRA FARG■■■■■■■■■■■■ON/OFF.
1110 PRINTTAB(26)"┌─────────┐"
1120 FORT=1TO5:PRINTTAB(26)"|         |":NEXT
1130 PRINTTAB(26)"└─────────┘":RETURN
1131 REM*********************************
1132 REM**  SPRITE NR 0. (CURSORN)    **
1133 REM*********************************
1140 V=53248
1150 POKEV+28,0
1160 POKE2040,200:FORD=64*200TO64*200+62:READS:POKED,S:NEXT:POKEV+39,1:RETURN
1170 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1180 DATA124,0,0,124,0,0,124,0,0,124,0,0,124,0,0,124,0,0,0,0,0,0,0
1190 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1197 PRINTTAB(26)"■H■ HOME"
1500 REM*********************************
1510 REM** PEEKPOSITIONER 21 STYCKEN **
1520 REM*********************************
2000 DIMW(21):FORI=1TO21:READW(I):NEXT:RETURN
2010 DATA1105,1145,1185,1225,1265,1305,1345,1385,1425,1465,1505,1545,1585,1625-
2020 DATA1665,1705,1745,1785,1825,1865,1905
2997 REM*********************************
2998 REM**MENY FöR PRINTNING AV DATA.**
2999 REM*********************************
3000 PRINT"■":POKEV,0:POKEV+1,0
3010 PRINT"■HUVUDSTYRD MENY FöR UTPRINTNING AV DATA."
3020 PRINT"■■"TAB(10)"■F1■ UT PRINTAT PA SKÄRM."
3030 PRINT"■■"TAB(10)"■F3■ UT PRINTAT PA PRINTER."
3040 PRINTTAB(24)"■■■■■■■■■■"
3050 PRINT"DATASATSER AV DENNA ■       ■"
3060 PRINT"SPRITE.■■■■■■■■■■■■■■       ■"
3070 PRINTTAB(24)"■"
3080 GETA$:IFA$<"■"ORA$>"■"THEN3080
3090 IFA$="■"THEN3110
3100 IFA$="■"THEN3170
3101 REM*********************************
3102 REM**RUTIN FöR PRINTNING SKÄRM.**
3103 REM*********************************
3110 C=1
3120 FORJ=1TO3:PRINT"■■"
3130 FORK=1TO21:PRINTF(C):C=C+1:NEXT
3140 PRINT"■■TRYCK PÅ F1";:IFC>50THENPRINT".OBS SISTA RADEN."
3150 GETA$:IFA$<"■"ORA$>"■"THEN3150
3160 NEXTJ:GOTO3230
3161 REM*********************************
3162 REM**RUTIN FöR PRINTNING PRINTER**
3163 REM*********************************
3170 C=1
3180 OPEN4,4
3190 FORJ=1TO3
3200 FORK=1TO21:PRINT#4,F(C):C=C+1:NEXTK
3210 NEXTJ
3220 REM*********************************
3221 REM*** AVSLUTNING AV PRINTNING ***
3222 REM*********************************
3230 PRINT"■"
3240 PRINT"■■VILL DU SE DATASATSERNA IGEN (J/N)"
3250 GETA$:IFA$=""THEN3250
3260 IFA$<>"J"THENPRINT"■■■":GOTO20
3270 GOTO3000
3997 REM*********************************
3998 REM***        PR SPRITE.        ***
3999 REM*********************************
4000 H=0:FORR=64*201TO64*201+62:H=H+1:READF(H):POKER,F(H):NEXT:RETURN
4010 DATA0,0,0,7,224,0,31,248,0,63,252,0,60,124,0,112,30,0,112,14,0
4020 DATA230,64,0,233,72,254,232,72,252,238,120,248,233,8,248,233,8,252,230,8
4030 DATA254,224,0,0,240,14,0,112,30,0,60,124,0,63,252,0,31,248,0,7,224,0
4031 REM*********************************
4032 REM***        HUVUD MENY.       ***
4033 REM*********************************
4040 PRINT"■"TAB(10)"■HEAVY SPRITE EDITOR.":POKEV+2,0:POKEV+3,0:POKEV+21,0
4050 PRINT"■■■■■■■■1■■ SPRITE EDITOR."
4060 PRINT"■■■■■■■■2■■ MATA IN DATA,MANUELLT."
4070 PRINT"■■■■■■■■3■■ LÄSA IN EN SPRITE FRAN BAND."
4080 PRINT"■■■■■■■■4■■ BANDA IN SPRITE PA ETT BAND."
4090 PRINT"■■■■■■■■TRYCK PA RESPEKTIVE TANGENT."
4100 GETA$:IFA$<"1"ORA$>"4"THEN4100
```

Forts på sidan 23

Om FORTH – Program exempel

LÄR DIG PROGRAMMERA



Liksom i nummer 10 förra året har det blivit dags för några programlistningar.

Den här gången kan jag erbjuda en definition av ordet ROLL, ett ord som ingår i FIG-FORTH-standarden, samt ett stackvisningsprogram.

ROLL är ett ord som låter användaren manipulera med stacken även om det ligger fler än tre tal där. Om man exempelvis har 5 parametrar på stacken så kan man "rolla" fram en i taget med frasen

5 ROLL

Stackvisningen kan vara lämplig för "nybörjare" för att undersöka hur stacken bär sig åt vid olika kommandon (använd t ex stackvisningen för att se hur ROLL fungerar!). En nackdel med programmet är att det bara tar ett ord i taget, så om du vill skriva 5 ROLL måste du dela upp det i två steg. Försök gärna att ändra programmet så det klarar flera ord.

Det kan kanske vara en lämplig övning till nästa artikel, då jag skall försöka visa en lösning på det problemet. Nästa artikel kommer förutom det innehålla en rescension av Leo Brodies nya bok: "Thinking FORTH". Den tar upp FORTH som ett språk och en filosofi (!) för problemlösning.

Med det avslutar jag denna månadsdos av FORTH och hoppas att ni kommer att ha nytta av ROLL eller stackvisningen, men framför allt att ni inspireras till att skriva egna program i detta annorlunda men mycket användbara språk. □

Ola Johansson

Lista 1

```
SCR # 18
( FIG-FORTH:  ROLL
                              OJ-840615)
 FORTH DEFINITIONS

CODE ROLL    ( roll# -- )
   1 # LDA.  SETUP JSR.  XSAVE STX.
   N LDA.  SEC.  1 # SBC.  CS IF.
   NE IF.  TAY.  .A ASL.  CLC.
   XSAVE ADC.  TAX.  BOT LDA.  PHA.
   BOT 1+ LDA.  PHA.  BEGIN.  DEX.  DEX.
   BOT LDA.  SEC STA.  BOT 1+ LDA.
   SEC 1+ STA.  DEY.  EQ UNTIL.  PLA.
   PUT JMP.  THEN.  THEN.  XSAVE LDX.
   NEXT JMP.
END-CODE

;S
```

```
SCR # 118
( FIG-FORTH: ROLL
                              OJ-840615)
 Detta ord ingar som standard i FIG-
FORTH, men finns inte ursprungligen till
64-ans FORTH.

 Här definieras ROLL i assembler (givet-
vis för att det ska ga snabbt) och text-
en har lagts upp som den bör, dvs med
en frasindelning med 2 mellanslag.

 Ordet ROLL fungerar sa att man anger
hur manga tal man vill rotera pa stacken
och vid anropet av ROLL sa utförs detto.
Det tal man har skrivit som antal räknas
inte med i "rotationen".
 2 ROLL är samma sak som SWAP
 3 ROLL kan användas i stället för ROT

(Lägg märke till hur olika strukturer
används i definitionen, som tex IF THEN
och BEGIN UNTIL.)

 Subrutinen SETUP flyttar det antal tal
som anges av A till en arbetsarea N.
```

Forth

Lista 2

```
SCR # 36
( INTERAKTIV STACKVISNING        OJ-841104)

FORTH
FORGET TASK  : TASK ;
6 USER S0

: .RGT   29 EMIT  ;
: .LFT   17 EMIT  ;                         ( -- *)
: .RVS   18 EMIT  ;                         ( -- *)
: .ROFF  146 EMIT ;                         ( -- *)
                                            ( -- *)
: CURPOS
   HOME  -DUP IF  0 DO      ( rad kol -- *)
   -DUP IF  0 DO  .LFT  LOOP  THEN  ;
  .RGT  LOOP  THEN
: .STACKPOS
   DUP 12 - ABS 29 CURPOS  .RVS   ( POS -- *)
   PICK  10 .R  .ROFF  ;

: DEPTH  SP@ S0 @ SWAP  - 2 /  ; ( -- *)

: SHOWSTACK
   DEPTH  10 >  7 ?ERROR  1064 440 32       ( -- *)
   FILL  DEPTH 1+ 1 DO    I .STACKPOS
   LOOP  ;
                                            -->
```

```
SCR # 136
( INTERAKTIV STACKVISNING        OJ-841104)

 Orden i denna och följande skärm ska
visa hur stacken paverkas vid olika
berakningar.

 CURPOS placerar markören pa önskad
plats pa skärmen.

 .STACKPOS skriver ut innehallet i en
stackcell pa en bestamd plats pa
skarmen.

 DEPTH ger stackens djup, som sedan an-
vands i SHOWSTACK for att visa hela
stackens innehall.

 Fortsättning i skärm 37 (137).
```

Lista 3

```
SCR # 37
( STACKVISNING FORTS.            OJ-841104)
( fortsättning från skärm 36)


: 0FIELD
   20 0 CURPOS  40 SPACES  ;                ( -- *)

: DOIT
   -FIND IF  DROP  CFA EXECUTE  ?STACK      ( -- *)
   ELSE  0 0 TIB @ 1-  (NUMBER)  DROP
   -DUP  DROP  THEN  ;

: COMMAND
   0FIELD  20 0 CURPOS  QUERY  DOIT  ;      ( -- *)

: HEADER
   PAGE  0 29 CURPOS ." Stacken:"  19 0     ( -- *)
    CURPOS ." Ditt kommando (avsluta med
 QUIT):"  HOME ." OLA'S STACKVISNING"  ;

: VISNING
   HEADER  BEGIN  SHOWSTACK  COMMAND        ( -- *)
   AGAIN  ;

        ( Ola Johansson 1984 )
```

```
SCR # 137
( STACKVISNING forts.            OJ-841104)

 Fortsättning från skärm 36 (136).

 0FIELD 'suddar ut' det tidigare
kommandot/talet.

 DOIT används i COMMAND för att utföra
det senaste kommandot. OBS att den bara
läser första ordet i kommandot. övrigt
ignoreras helt.

 VISNING är huvuddefinitionen, som ut-
för hela stackvisningen. Den avbryts
genom att man skriver ordet QUIT.


              / Ola Johansson 1984  (C)
```

Forts från sidan 14

"COMAL-SKOLAN"

Sen var det inte mer i den här FOR-snurrans verksamhetslista! Den avslutas med satsen:

ENDFOR I

Nu kan vi lägga in alla DATA-satserna med tillhörande tonvärden. Se programlistan.

Sist avslutas med procedurens slutrad:

ENDPROC LJUDINITIERING

PROCEDUR FÖR OMSLAGSKONTROLL

Vi vill ju också kunna lägga in värden för ADSR, dvs Attack, Decay, Sustain och Release, tonens omslagskurva.

Den ser ut som på en bild, som fanns införd i VIC rapport nr 3 årgång 2. Vi återger bilden här i något renritad form som visar hur denna funktion ser ut i SID-chipet.

Här behöver vi en längre parameterlista, som låter oss sätta värden separat för stämma, attack, decay, sustain och release. Vi får följande procedurhuvud:

PROC OMSLAG(STÄMMA,STIGTID,SJUNKNING,HÅLLNIVÅ,AVKLINGNING)

För att "rikta" in oss på rätt stämma lägger vi i en tilldelningssats in ett värde på en variabel "J", som bara används inom proceduren.

J:=SID+(STÄMMA-1)*7

Denna tilldelning hämtar värdet på STÄMMA och låter J få rätt värde beroende på om man väljer stämma 1, 2 eller 3.

Själva tilldelningen av värden på omslagsvariablerna ADSR sker i två satser:

POKE J+5, STIGTID*16+SJUNKNING
POKE J+6, HÅLLNIVÅ*16+AVKLINGNING

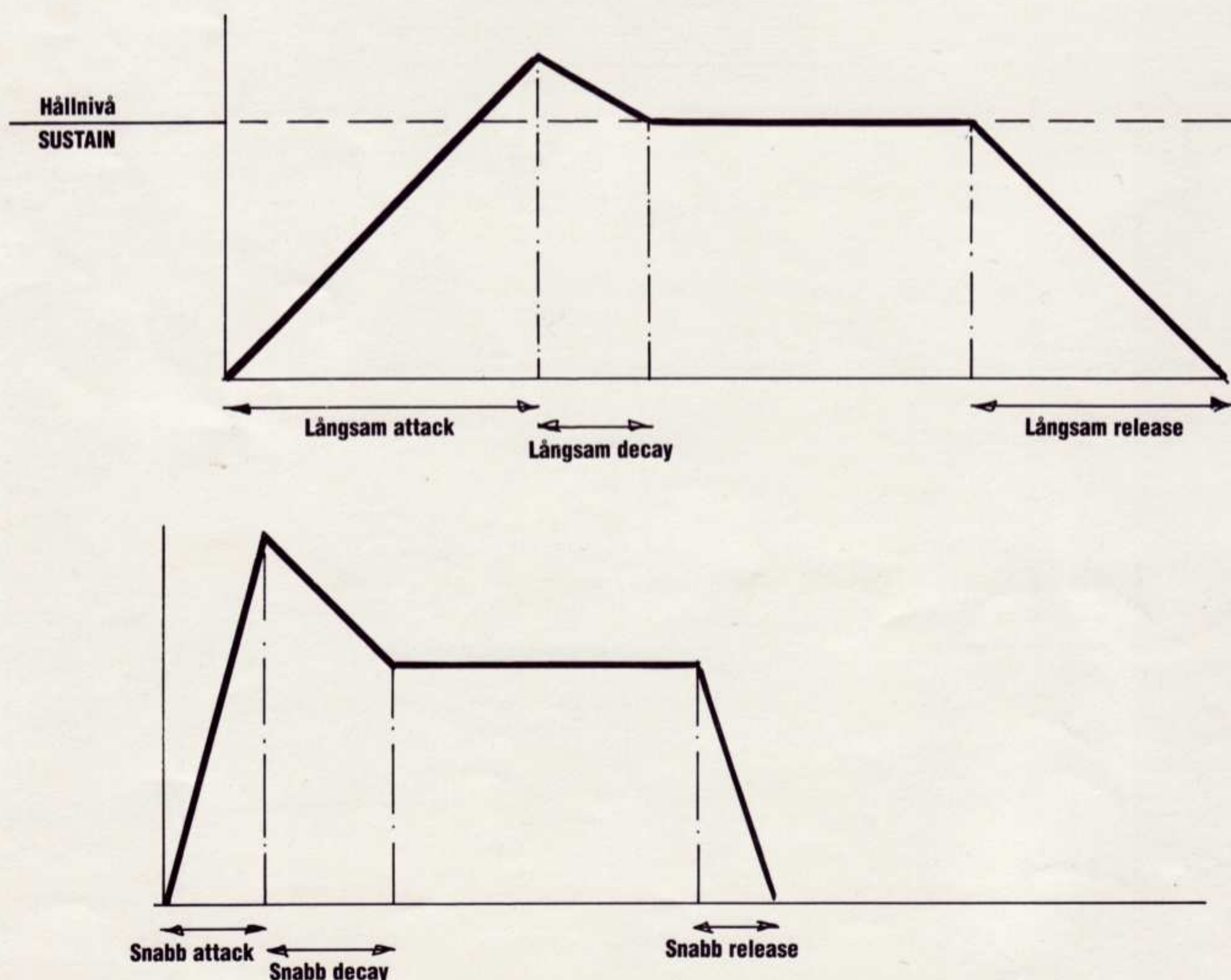
och proceduren avslutas med

ENDPROC OMSLAG

Se även programlistningarna.

Comalskolan fortsätter i nästa nummer av VIC rapport.

Å Fredriksson



Forts "SPRITE EDITOR"

```
4110 Q=VAL(A$):PRINT"▒▒▒"
4120 ONQGOTO20,4140,4340,4220
4130 REM**********************************
4131 REM***    MANUELL INMATNING.    ***
4132 REM**********************************
4140 PRINT"▒"TAB(12)"▒MANUELL INMATNING"
4145 C=0
4150 FORJ=64*201TO64*201+62:C=C+1
4160 PRINT"▒▒▒▒INMATNINGS NUMMER";C
4170 INPUT"▒▒DATA ?";F(C):IFF(C)<0ORF(C)>255THEN4170
4180 POKEJ,F(C):NEXT
4190 PRINT"▒▒▒NU ÄR DET KLART...TRYCK PÅ EN TANGENT..."
4200 GETA$:IFA$=""THEN4200
4210 C=0:GOTO4040
4211 REM**********************************
4212 REM***          BANDNING.        ***
4213 REM**********************************
4220 PRINT"▒"TAB(11)"▒BANDNING AV SPRITE."
4230 PRINT"▒▒▒DU HAR INTE ANGRAT DIG..............(J/N)
4240 GETA$:IFA$=""THEN4240
4250 IFA$<>"N"THEN4040
4270 OPEN1,1,1
4280 FORJ=0TO63
4290 PRINT#1,F(J)
4300 NEXT:CLOSE1
4310 PRINT"▒▒NU ÄR DET KLART.....TRYCK PÅ EN TANGENT."
4320 GETA$:IFA$=""THEN4320
4330 GOTO4040
4331 REM**********************************
4332 REM***  INLÄSNING AV SPRITE.   ***
4333 REM**********************************
4340 PRINT"▒"TAB(10)"▒INLÄSNING AV SPRITE."
4350 PRINT"▒▒▒DU HAR INTE ANGRAT DIG..............(J/N)
4360 GETA$:IFA$=""THEN4360
4370 IFA$<>"N"THEN4040
4390 OPEN1,1,0
4400 FORJ=0TO63
4410 INPUT#1,F(J)
4420 NEXT
4430 C=0
4440 FORJ=64*201TO64*201+62:C=C+1:POKEJ,F(C):NEXTJ
4450 PRINT"▒▒NU ÄR DET KLART.....TRYCK PÅ EN TANGENT."
4460 GETA$:IFA$=""THEN4320
4470 GOTO4040

READY.
```

□



VIRUS – VIC 20

Ett adventurespel på svenska. Kan det vara något? Här kommer en listning till en 16k textadventure i sjukhusmiljö för VIC 20. Nedan följer en liten beskrivning av möjligheterna i VIRUS.

När Du startar programmet så kommer frågan

'VILL DU LADDA IN SPARAT SPEL?'

Om det är första gången Du kör VIRUS så svarar Du NEJ och ditt äventyr börjar. I annat fall kanske Du har utnyttjat möjligheten att spara det spel Du håller på med för tillfället (se nedan). En finess som kan vara bra att ha eftersom spelet tar ett par timmar. Ett JA laddar in spelet och Du kan fortsätta i handlingen där Du slutade.

När Du väl har kommit förbi den här punkten så skrivs lite krafs ut och Du får sedan frågan

'VAD SKALL JAG GÖRA NU?'

Nu kan Du tala om för Din Gubbe vart han skall gå, vad han ska göra etc på det lilla sjukhuset. Följande kommandon bör man ha kännedom om, men observera att det finns åtskilliga ord men att man får komma på dessa själv.

K eller KARTA

Visar en karta över sjukhuset.

T eller TITTA

Talar om vad din gubbe ser.

Ö eller ÖST (Pundtecken om Du ej har åäö)

Gubben går mot öster.

På samma sätt kan man gå i andra riktningar. Vilka riktningar som är möjliga framgår av utskrift vid TITTA kommandot. OBSERVERA dock att dörrar etc med fördel öppnas innan man går igenom dem.

U eller UTRUSTNING

Visar vilka prylar du släpar med dig.

Som vi såg ovan så går det att spara sitt spel så att man inte behöver börja om från början igen nästa gång man ska spela VIRUS. Detta gör man genom att stoppa in en blank kassett och spola tillbaks denna till början. Man skriver sedan

SPAR SPEL

och spelet sparas.

Nästa gång Du startar upp så stoppar Du sedan in denna kasset och svarar JA på den inledningsfråga som vi diskuterade ovan.

Om Du vill avsluta utan att spara spelet så kan Du ge ett av följande kommandon:

END
SLUTA
STOP

och programmet avslutas.

VAD GÅR VIRUS UT PÅ?

Din uppgift är att rädda de fem patienterna som har olika sjukdomar. Tänk på att många sjukdomar är smittsamma och att läkare normalt undersöker...

Man bör kanske inte heller låta avlidna Patienter (om detta nu skulle hända) ligga och skräpa. Otrevliga lukter kan ju uppstå...

Och snälla du, låt inte patienterna lida allt för mycket! Nästa gång så är det kanske du som befinner dig i deras situation.

OBS! Programmet kräver att man har 16K expansionsminne.

LYCKA TILL DOKTORN! □

Å Hedman

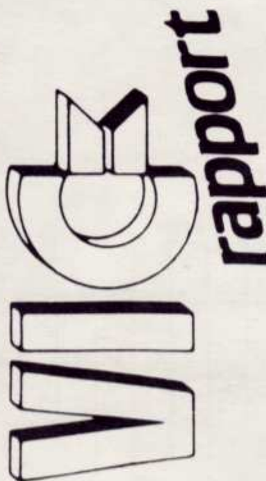
Program "VIRUS"



```
READY.

11 PRINT"█        ***X\_/****":PRINT:PRINT
12 PRINT"█U AR DEN ENDE LAKARENPA ETT SJUKHUS MED SMARESURSER.
15 PRINT" HJALP DE STACKARS      PATIENTERNA.":PRINT:PRINT:PRINT
20 PR%=0:XX%=0
30 PRINT"◣XILL DU LADDA":PRINT"SPARAT PROGRAM.":INPUTA$
40 IF LEFT$(A$,1)="J"THENXX%=10:GOTO100
50 IF LEFT$(A$,1)="N"THEN 100
60 PRINT:PRINT"XAD MENAR DU MED ";A$;"?":PRINT:PRINT:PRINT"'AG FRAGADE":GOTO30
100 :
120 DIMJ%(5,6)
122 DIMD%(9),I%(25)
130 FORI=1TO5
140 J%(I,0)=I
150 FORJ=1TO6
160 J%(I,J)=1
170 NEXTJ:NEXTI
172 PR%=1000
175 A1$="MARKT ":A2$="HYLLA ":A3$="BURK "
180 FORI=1TO5
190 RN%=INT(RND(TI)*4+1)
200 IFRN%=1THENJ%(I,3)=38:GOTO240
210 IFRN%=2THENJ%(I,3)=39:GOTO240
220 IFRN%=3THENJ%(I,3)=0 :GOTO240
230 J%(I,3)=35
240 NEXTI
250 J%(RN%,3)=37
290 FORI=1TO9
300 D%(I)=0:NEXTI
310 D%(1)=1:D%(4)=1
320 RI$="(((.(((/(((1(((2(((3(/)4.0*;/1(602+(13,(2(-(5(.((4(((70(689(7(:((((7(((
8((/("
```

Prenumerera på



Helårsprenumeration 140:–

Som prenumerant får du:
- VIC rapport direkt i brevlådan varje månad.
- Annonsera gratis i VIC rapport.
- Köpa kassetter billigare hos VIC kassettservice.
- mm

VIC kassettservice

Varje månad utkommer det en kassett parallellt med VIC rapport som innehåller de program som finns listade i tidningen.

T-shirts med VIC motiv

Du kan även beställa VIC t-shirts via VIC rapport.

Använd postgirokortet på andra sidan av detta insticksblad och fyll i dina önskemål.

VIC rapport årg 2 nr 3	15:–
VIC rapport årg 2 nr 4	15:–
VIC rapport årg 2 nr 5/6	20:–
VIC rapport årg 3 nr 1	20:–
VIC rapport årg 3 nr 2	15:–
VIC rapport årg 3 nr 3	15:–
VIC rapport årg 3 nr 4	15:–
VIC rapport årg 3 nr 5/6	20:–
VIC rapport årg 3 nr 7	15:–
VIC rapport årg 3 nr 8	SLUT
VIC rapport årg 3 nr 9	SLUT
VIC rapport årg 3 nr 10	15:–
VIC rapport årg 3 nr 11/12	20:–
VIC rapport 1985 nr 1	17:50
VIC rapport 1985 nr 2	17:50
VIC rapport 1985 nr 3	17:50
VIC rapport 1985 nr 4	17:50
VIC rapport 1985 nr 5/6	25:–

VIC kassettservice	Prenumerant	Icke prenumerant
☐ 3 nr	100:–	125:–
☐ 6 nr	190:–	215:–
☐ 10 nr	300:–	325:–

Porto

1 tidning	7:00	4–7 tidningar	14:–
2–3 tidningar	10:50	>7 tidningar	21:50

Avskiljes före inbetalning



POSTGIROT SVERIGE — **INBETALNING / GIRERING B** 2 • Konto • Avg • Bel •

Meddelande till betalningsmottagaren

Jag beställer:
En årsprenumeration på VIC rapport 140:–
VIC rapport årg ______ nr ______ a ______
VIC rapport årg ______ nr ______ a ______
VIC rapport årg ______ nr ______ a ______
VIC rapport årg ______ nr ______ a ______
VIC rapport årg ______ nr ______ a ______
VIC kassettservice nr ______ a ______

T-shirt med VIC motiv a 49:–
.......st typ 1 2 storlek S M ______

+Porto

Summa kronor ______

Till postgirokonto nr: **8 46 46-9** | Avgift | Kassastämpel

Betalningsmottagare (endast namn): **HANDIC PRESS**

Avsändare (namn och postadress)

Eget kontonr vid girering

Svenska kronor | öre

Bl 2021.04 S (dec 82) 35067 84.12

I DETTA FÄLT FÅR ANTECKNINGAR INTE GÖRAS – RESERVERAT FÖR POSTGIROT

846469#02#

GRATISANNONS för VIC prenumeranter

Jag är prenumerant på VIC rapport och vill ha gratisannons enligt nedanstående manuskript införd snarast i VIC rapport.

Namn: ....

Adress: ....

Postadress: ....

Prenumerationsnummer: ....

Rubrik:

Sändes till:
VIC rapport, Redaktionen,
Box 42054, 126 12 Stockholm

SÄND IN DIN BESTÄLLNING NU!

I FÄLTET OVAN FÅR ANTECKNINGAR INTE GÖRAS – RESERVERAT FÖR POSTGIROT

Meddelanden till betalningsmottagaren kan inte lämnas på denna sida

Med inbetalningskort kan betalning ske till postgiro- och personkonton. Det tas emot av postkontor, poställen, lantbrevbärare, PKbankskontor och sparbanker.

Inbetalningskort kan även användas för girering från postgiro- och personkonton samt för betalning via PKbankens banklönegiro eller sparbanksgirot.

Det går också bra att använda inbetalningskortet på posten i Danmark, Finland och Norge eller för girering från postgirokonton i dessa länder.

Blanketten skall skrivas ut med bläck, kulpenna eller på maskinell väg. Man får inte radera, stryka över eller göra annan ändring av tryckt eller skriven text. Angivet kontonummer eller belopp får inte ändras.

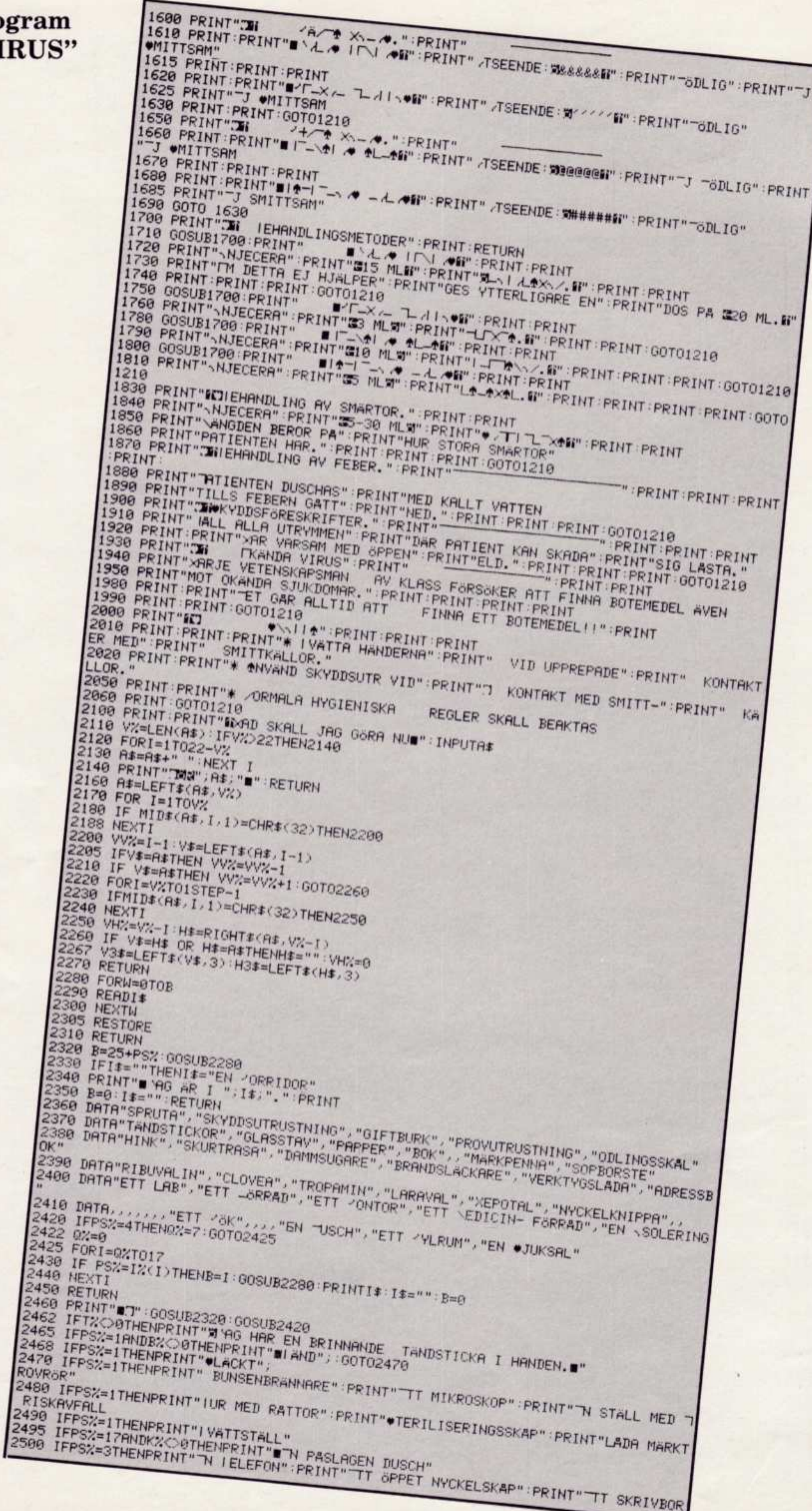
DAVID BROBERG, STHLM

Program "VIRUS"

```
330 FORI=1TO5
340 J%(I,2)=19
350 NEXTI
355 PS%=6:B%=0:S%=0
360 FORI=0TO7:I%(I)=4:NEXTI:I%(5)=1
365 I%(22)=50:I%(7)=14
370 FORI=18TO21:I%(I)=4:NEXTI
380 FORI=8TO10:I%(I)=1:NEXTI:I%(17)=3
385 I%(23)=3:I%(9)=100
387 J%(3,5)=2:J%(4,6)=0
390 FORI=11TO16:I%(I)=2:NEXTI:I%(22)=50
400 IF XX%<>0THENGOSUB30000
500 GOSUB3050:GOSUB3130:GOSUB6100:GOSUB6400:GOSUB7700:GOSUB7800:GOSUB8000:GOSUB9
000
520 GOSUB3000:GOSUB2800:GOSUB4080:GOSUB6560:GOSUB10000
550 GOSUB2100:GOSUB2160:N%=N%+1
560 IFV$="END"ORV$="STOP"ORV$="SLUTA"THEN3170
565 IFV3$="KAR"ORA$="K"THEN1000
570 IFV3$="TIT"ORA$="T"THEN2460
575 IFV3$="HJä"THEN3180
580 IFV3$="ANV"THEN3190
590 IFV3$="NOR"ORA$="N"ORV$="SYD"ORA$="S"ORLEFT$(V$,3)="VäS"ORA$="V"THEN3200
595 IFV$="öST"ORA$="ö"THEN3200
600 IFV$="GÅ"ANDH$<>""THENV$=H$:A$=H$:GOTO590
610 IFV3$="öPP"THEN3400
620 IFV3$="VäC"THEN4040
630 IFV3$="STä"THEN4100
640 IFV$="U"ORV3$="UTR"THEN4200
650 IFLEFT$(V$,2)="TA"THEN4350
660 IFV3$="SLä"ORV3$="LäM"THEN4600
670 IFV3$="LåS"ANDH3$="DöR"THEN4800
675 IFV3$="LäS"ANDH3$="JOU"THEN5000
680 IFV3$="LAG"THEN5050
690 IFV3$="RIN"THEN5100
700 IFV3$="LäS"THEN5300
710 IFV3$="BLä"THEN5720
720 IFV3$="SIT"THENPRINT"■'AG SER INGET ATT     SITTA PÅ.":GOTO999
730 IFV3$="SåG"THEN5800
740 IFV3$="TRY"THEN5900
750 IFV3$="TVä"THEN6300
760 IFV3$="TäN"THEN6500
770 IFV3$="UND"THEN6700
780 IFV3$="FYL"THEN7300
790 IFV3$="INJ"ORV$="GE"THEN7500
800 IFV3$="SPA"THEN20000
850 IFA$<>""THEN998
996 PRINT"■/äGOT BöR JAG VäL I   ALLA FALL GöRA.":V%=FRE(8):GOTO500
998 PRINT"■VET EJ HUR MAN ";A$:V%=FRE(8):A$="":GOTO500
999 A$="":V$="":V3$="":H$="":H3$="":PRINT"╓┬/":V%=FRE(8):PRINT:GOTO500
1000 POKE36879,109:PRINT"┐■";TAB(8);"▒   ▒";CHR$(142)
1010 PRINT"  LAB   ▒   ▒  KöK
1020 PRINTTAB(12);"▒
1030 PRINTTAB(8);"▒    ▒▒ ▒▒▒▒▒▒▒
1040 PRINT"┐▒▒▒▒▒▒▒▒▒▒▒
1050 PRINTTAB(8);"▒    ▒▒▒▒▒▒▒▒▒▒
1060 PRINT" STäD-  ▒
1070 PRINT" FöRRAD ▒    ▒▒▒▒▒▒▒▒ S
1080 PRINTTAB(8);"▒         ▒ J
1090 PRINT"▒▒▒▒▒▒▒▒▒▒▒       ▒ U
1100 PRINTTAB(8);"▒  ▒▒▒▒▒   ▒ K
1110 PRINT" KONTOR    ▒ D ▒   ▒ *
1120 PRINTTAB(8);"▒  ▒ U ▒   ▒ S
1130 PRINT"▒▒▒▒▒▒▒▒▒▒▒  ▒ S    ▒ A
1140 PRINTTAB(8);"▒  ▒ C ▒   ▒ L
1150 PRINT" MEDICIN▒  ▒ H ▒   ▒
1160 PRINT" FöRRAD    ▒▒▒▒▒   ▒
1170 PRINTTAB(8);"▒  ▒ K ▒   ▒
1180 PRINT"▒▒▒▒▒▒▒▒▒▒▒  ▒ Y    ▒
1190 PRINT" ISOLER-▒  ▒ L ▒   ▒
1200 PRINT" ING       ▒   ▒   ▒
1210 PRINT"▒TRYCK !!!           ▒
1220 GETA$:IF A$=""THEN1220
1230 PRINT"┐■"CHR$(14):POKE36879,25:GOTO999
1250 PRINT"▒▒┐     |┐└‾┘┌/└┐●|↑     "
1260 PRINT:PRINT"\AMMA&┐APPA      154086
1270 PRINT"‾VA........       146080
1280 PRINT"‾JURPARKEN.       420000
1290 PRINT"↑POTEKET...       180246
1300 PRINT"┘UNE.......       112213
1310 PRINT"FAM /RUT...       203410
1320 PRINT"■":PRINT:PRINT:GOTO1210
1330 PRINT"┐▒ \NNEHåLLSFöRTECKNING":PRINT" ──────────────────────":PRINT"●ID:
1340 PRINT"1  ┌DLING.
1350 PRINT"2  /äNDA VIRUS.
1360 PRINT"3  /äNDA VIRUS.
1370 PRINT"4  |EHANDLINGSM.
1380 PRINT"5  |EHANDLINGSM.
1390 PRINT"6  |EHANDLINGSM.
1400 PRINT"7  |EHANDLINGSM.
1450 PRINT"8  |EH. AV SMäRTOR.
1460 PRINT"9  |EH. AV FEBER.
1470 PRINT"10 ●KYDDSFöRESKRIFTER."
1475 PRINT"┐11 ┌KäNDA VIRUS.
1480 PRINT"12 ●MITTA.":PRINT:PRINT:PRINT
1490 GOTO1210
1500 PRINT"┐SID 1▒
1510 PRINT:PRINT"  ┌DLING AV VIRUS.":PRINT" ──────────────":PRINT
1520 PRINT"* |AG UTRUSTNING":PRINT"  (ODLINGSSKåL OCH":PRINT"  PROVUTRUSTNING)."
:PRINT
1570 PRINT"* |AG PROV AV PATIENT.":PRINT
1580 PRINT"* /NDERSöK I MIKRO-":PRINT"  SKOP.":PRINT:PRINT:PRINT:PRINT
1590 PRINT"|+/⁄ ┐| ●\\||┌─┐●/‾/.":PRINT:PRINT:GOTO1210
```

▶

Program "VIRUS"



```
1600 PRINT"■     ⁄Å⁄⌐♠ ×⌐–♠.":PRINT"          ────────
1610 PRINT:PRINT"■⌐⅄♠ |⌐\| ♠■":PRINT"⁄TSEENDE:■&&&&&■":PRINT"¬ÖDLIG":PRINT"¬J
●MITTSAM"
1615 PRINT:PRINT:PRINT
1620 PRINT:PRINT"■⁄Γ–×⁄– ⅂⅄||⌐●■":PRINT"⁄TSEENDE:■⁄⁄⁄⁄⁄■":PRINT"¬ÖDLIG"
1625 PRINT"¬J ●MITTSAM
1630 PRINT:PRINT:GOTO1210
1650 PRINT"■     ⁄+⁄⌐♠ ×⌐–♠.":PRINT"          ────────
1660 PRINT:PRINT"■|Γ–\♠| ♠ ♠L–♠■":PRINT"⁄TSEENDE:■@@@@@■":PRINT"¬J ¬ÖDLIG":PRINT
"¬J ●MITTSAM
1670 PRINT:PRINT:PRINT
1680 PRINT:PRINT"■|♠–|¬–⌐ ♠ –⅄ ♠■":PRINT"⁄TSEENDE:■######■":PRINT"¬ÖDLIG"
1685 PRINT"¬J SMITTSAM"
1690 GOTO 1630
1700 PRINT"■  |EHANDLINGSMETODER":PRINT:RETURN
1710 GOSUB1700:PRINT"      ■⌐⅄♠ |⌐\| ♠■":PRINT:PRINT
1720 PRINT"⌐NJECERA":PRINT"■15 ML■":PRINT"■L⌐|⅄♠×⌐⁄.■":PRINT:PRINT
1730 PRINT"ΓM DETTA EJ HJÄLPER":PRINT"GES YTTERLIGARE EN":PRINT"DOS PÅ ■20 ML.■"
1740 PRINT:PRINT:PRINT:GOTO1210
1750 GOSUB1700:PRINT"      ■⁄Γ–×⁄– ⅂⅄||⌐●■":PRINT:PRINT
1760 PRINT"⌐NJECERA":PRINT"■3 ML■":PRINT"–LΓ×♠.■":PRINT:PRINT:GOTO1210
1780 GOSUB1700:PRINT"      ■|Γ–\♠| ♠ ♠L–♠■":PRINT:PRINT
1790 PRINT"⌐NJECERA":PRINT"■10 ML■":PRINT"|–Γ♠\⁄.■":PRINT:PRINT:PRINT:GOTO1210
1800 GOSUB1700:PRINT"      ■|♠–|¬–⌐ ♠ –⅄ ♠■":PRINT:PRINT
1810 PRINT"⌐NJECERA":PRINT"■5 ML■":PRINT"L♠–♠×♠L.■":PRINT:PRINT:PRINT:PRINT:GOTO
1210
1830 PRINT"■■|EHANDLING AV SMÄRTOR.":PRINT:PRINT
1840 PRINT"⌐NJECERA":PRINT"■5-30 ML■":PRINT"●⁄T|⅂¬×♠■":PRINT:PRINT
1850 PRINT"\ÄNGDEN BEROR PÅ":PRINT"HUR STORA SMÄRTOR"
1860 PRINT"PATIENTEN HAR.":PRINT:PRINT:PRINT:GOTO1210
1870 PRINT"■■|EHANDLING AV FEBER.":PRINT"──────────────":PRINT:PRINT:PRINT
:PRINT
1880 PRINT"¬ATIENTEN DUSCHAS":PRINT"MED KALLT VATTEN
1890 PRINT"TILLS FEBERN GÅTT":PRINT"NED.":PRINT:PRINT:PRINT:GOTO1210
1900 PRINT"■■KYDDSFÖRESKRIFTER.":PRINT"──────────────":PRINT:PRINT:PRINT
1910 PRINT"|ÅLL ALLA UTRYMMEN":PRINT"DÄR PATIENT KAN SKADA":PRINT"SIG LÅSTA."
1920 PRINT:PRINT"×AR VARSAM MED ÖPPEN":PRINT"ELD.":PRINT:PRINT:PRINT:GOTO1210
1930 PRINT"■     ΓKÄNDA VIRUS":PRINT"──────────":PRINT:PRINT
1940 PRINT"×ARJE VETENSKAPSMAN   AV KLASS FÖRSÖKER ATT FINNA BOTEMEDEL ÄVEN
1950 PRINT"MOT OKÄNDA SJUKDOMAR.":PRINT:PRINT:PRINT:PRINT
1980 PRINT:PRINT"¬ET GÅR ALLTID ATT    FINNA ETT BOTEMEDEL!!":PRINT
1990 PRINT:PRINT:GOTO1210
2000 PRINT"■■          ●⌐||♠":PRINT:PRINT:PRINT
2010 PRINT:PRINT:PRINT"* |VÄTTA HÄNDERNA":PRINT"  VID UPPREPADE":PRINT"  KONTAKT
ER MED":PRINT"  SMITTKALLOR."
2020 PRINT:PRINT"* ♠NVÄND SKYDDSUTR VID":PRINT"⅂ KONTAKT MED SMITT-":PRINT"  KÄ
LLOR."
2050 PRINT:PRINT"* ⁄ORMALA HYGIENISKA    REGLER SKALL BEAKTAS
2060 PRINT:GOTO1210
2100 PRINT:PRINT"■×AD SKALL JAG GÖRA NU■":INPUTA$
2110 V%=LEN(A$):IFV%>22THEN2140
2120 FORI=1TO22-V%
2130 A$=A$+" ":NEXT I
2140 PRINT"■■";A$;"■":RETURN
2160 A$=LEFT$(A$,V%)
2170 FOR I=1TOV%
2180 IF MID$(A$,I,1)=CHR$(32)THEN2200
2188 NEXTI
2200 VV%=I-1:V$=LEFT$(A$,I-1)
2205 IFV$=A$THEN VV%=VV%-1
2210 IF V$=A$THEN VV%=VV%+1:GOTO2260
2220 FORI=V%TO1STEP-1
2230 IFMID$(A$,I,1)=CHR$(32)THEN2250
2240 NEXTI
2250 VH%=V%-I:H$=RIGHT$(A$,V%-I)
2260 IF V$=H$ OR H$=A$THENH$="":VH%=0
2267 V3$=LEFT$(V$,3):H3$=LEFT$(H$,3)
2270 RETURN
2280 FORW=0TOB
2290 READI$
2300 NEXTW
2305 RESTORE
2310 RETURN
2320 B=25+PS%:GOSUB2280
2330 IFI$=""THENI$="EN ⁄ORRIDOR"
2340 PRINT"■⌐AG ÄR I ";I$;".":PRINT
2350 B=0:I$="":RETURN
2360 DATA"SPRUTA","SKYDDSUTRUSTNING","GIFTBURK","PROVUTRUSTNING","ODLINGSSKÅL"
2370 DATA"TÄNDSTICKOR","GLASSTAV","PAPPER","BOK",,"MÄRKPENNA","SOPBORSTE"
2380 DATA"HINK","SKURTRASA","DAMMSUGARE","BRANDSLÄCKARE","VERKTYGSLÅDA","ADRESSB
OK"
2390 DATA"RIBUVALIN","CLOVEA","TROPAMIN","LARAVAL","XEPOTAL","NYCKELKNIPPA",,
2400 DATA"ETT LAB","ETT –ÖRRÅD","ETT ⁄ONTOR","ETT \EDICIN- FÖRRÅD","EN ⌐SOLERING
"
2410 DATA,,,,,,,"ETT ⁄ÖK",,,,"EN ¬USCH","ETT ⁄YLRUM","EN ●JUKSAL"
2420 IFPS%=4THENQ%=7:GOTO2425
2422 Q%=0
2425 FORI=Q%TO17
2430 IF PS%=I%(I)THENB=I:GOSUB2280:PRINTI$:I$="":B=0
2440 NEXTI
2450 RETURN
2460 PRINT"■⅂":GOSUB2320:GOSUB2420
2462 IFT%<>0THENPRINT"■⌐AG HAR EN BRINNANDE  TÄNDSTICKA I HANDEN.■"
2465 IFPS%=1ANDB%<>0THENPRINT"■|ÄND";:GOTO2470
2468 IFPS%=1THENPRINT"●LACKT";
2470 IFPS%=1THENPRINT" BUNSENBRÄNNARE":PRINT"¬TT MIKROSKOP":PRINT"¬N STÄLL MED ¬
ROVRÖR"
2480 IFPS%=1THENPRINT"|UR MED RÅTTOR":PRINT"●TERILISERINGSSKÅP":PRINT"LÅDA MÄRKT
 RISKAVFALL
2490 IFPS%=1THENPRINT"|VÄTTSTÄLL"
2495 IFPS%=17ANDK%<>0THENPRINT"■¬N PÅSLAGEN DUSCH"
2500 IFPS%=3THENPRINT"¬N |ELEFON":PRINT"¬TT ÖPPET NYCKELSKÅP":PRINT"¬TT SKRIVBOR
```



```
D"
2510 IFPS%=3THENPRINT"¬N ♥KRIVMASKIN"
2520 IFPS%=3ANDD%(0)<>2THENPRINT"¬TT ♥TÄNGT `OURNALSKAP"
2530 IFPS%=3ANDD%(0)=2THENPRINT"¬TT öPPET `OURNALSKAP"
2540 IFPS%=4THENPRINT"¬N LADA ♥PRUTOR":PRINT"♥KYDDSUTRUSTNINGAR"
2545 IFPS%=4THENPRINT"_LERA BURKAR MARKTA  ⁄IFT⁄"
2550 IFPS%=4THENPRINT"‾ROVUTRUSTNING":PRINT"┌DLINGSSKALAR":IFI%(23)=4THENPRINT"⁄
YCKELKNIPPA"
2560 IFPS%=4THENPRINT"¬N HYLLA MED MEDICIN"
2565 RESTORE
2620 IFPS%=13THENPRINT"‾IVERSE KöKSUTRUSTNING"
2630 IFPS%=13ANDS%=0THENPRINT"♥OVANDE";
2640 IFPS%=13ANDS%=1THENPRINT"♠RBETANDE";
2650 IFPS%=13THENPRINT"⁄öKSPERSONAL"
2660 IFPS%=17THENPRINT"¬N ▀LöD■ KNAPP":PRINT"¬N ▀ILA■ KNAPP"
2670 IFPS%=18THENPRINT"‾ET äR VäLDIGT KALLT  HäR INNE"
2680 IFPS%=19THENPRINT"¬TT TVATTSTALL"
2690 IFD$<>""THENPRINTD$
2691 FORI=1TO5
2692 IFJ%(I,2)=PS%THENPRINT"PATIENT:";I
2695 NEXTI
2700 IFV%(0)<>0THENPRINT"▀⁄ORD ";
2710 IFV%(1)<>0THENPRINT"▀♥YD ";
2720 IFV%(2)<>0THENPRINT"▀×äST ";
2730 IFV%(3)<>0THENPRINT"▀öST";
2735 PRINT" "
2740 GOTO999
2800 D0%=0:D1%=0:D$=""
2810 IFPS%=7THEN2880
2820 FORI=6TO12
2825 IFI<8ANDPS%=ITHEND0%=D%(I-5):D=I-5:GOTO2960
2830 IFI>8ANDPS%=ITHEND0%=D%(I-6):D=I-6:GOTO2960
2840 NEXTI
2850 IFPS%=15THEND0%=D%(8):D=8:GOTO2960
2860 IFPS%=16THEND0%=D%(9):D=9:GOTO2960
2870 GOTO2990
2880 D0%=D%(2):D1%=D%(7):D=2
2890 IFD0%<>2THEND$="♥TäNGD "
2900 IFD0%=2THEND$="öPPEN  "
2910 D$=D$+"DöRR I VäSTER  "
2920 IFD1%<>2THEND$=D$+"♥TäNGD "
2930 IFD1%=2THEND$=D$+"öPPEN  "
2940 D$=D$+"DöRR I öSTER"
2950 GOTO2990
2960 IFD0%<>2THEND$="STäNGD "
2970 IFD0%=2 THEND$="öPPEN  "
2980 D$=D$+"DöRR"
2990 RETURN
3000 FORI=0TO3
3010 V%(I)=ASC(MID$(RI$,4*PS%+I-3,1))-40
3030 NEXTI
3040 RETURN
3050 IFL%=1ANDD%(9)=2THEN3070
3055 IFPS%=16ORPS%=18THENRETURN
3060 IFPS%=14ORPS%=15ORPS%=17THENRETURN
3070 IFD%(9)=1ORD%(9)=0THENL%=0:RETURN
3075 L%=1
3080 PRINT"▀▀_♠_\!!!!!!               ":PRINT"▀▀⁄ IL_^♥¬_ ‖TT⁄!"
3090 POKE36878,15
3092 FORL=1TO10
3094 FORM=180TO235STEP2
3100 POKE36876,M:NEXTM:NEXTL
3102 FORM=1TO10
3110 POKE36876,0:NEXTM
3112 FORM=1TO100:NEXTM
3120 POKE36878,0:RETURN
3130 IFPS%=1THENRETURN
3140 IFB%<>1THENRETURN
3150 PRINT"▀▀_♠_\!!!!      ":PRINT"▀▀I_♠⁄‾ \ L♠!!":PRINT:PRINT"■¬LDEN SPRIDER SIG
.":PRINT
3170 PRINT"▀▀♠‾×⁄I ⁄‾ ♥L⁄!!!!!■":END
3180 PRINT"■×EM SKALL HJäLPA VEM   EGENTLIGEN????":GOTO999
3190 PRINT"■×AR SNALL OCH UTTRYCK  DIG Pä ANNAT SäTT.":GOTO999
3200 IFV3$="NOR"ORA$="N"THENI=0
3210 IFV$="SYD"ORA$="S"THENI=1
3220 IFV3$="VäS"ORA$="V"THENI=2
3230 IFV$="öST"ORA$="ö"THENI=3
3240 IFV%(I)=0THENPRINT"⁄AN INTE GA I DEN        RIKTNINGEN.":GOTO999
3250 IFPS%<>7THEN3270
3255 IFI=2ANDD%(2)<>2THEN4000
3260 IFI=3ANDD%(7)<>2THEN4000
3265 PS%=V%(I):GOTO999
3270 IFPS%<6ORPS%>11THEN 3300
3280 IFI=2ANDD%(D)<>2THEN4000
3290 GOTO3265
3300 IFPS%=12ANDD%(6)<>2ANDI=0THEN4000
3310 IFPS%=15ORPS%=16THEN3330
3320 GOTO3265
3330 IFI=2ANDD%(D)<>2THEN4000
3340 GOTO3265
3400 IFH3$="JOU"THEN3700
3405 IFH3$<>"DöR"THENPRINT"■×AD SKALL JAG GöRA????":GOTO999
3407 IFPS%=8THENC$="NAGON DöRR":GOTO4020
3410 IFPS%<>7THEN3500
3420 PRINT"×ILKEN DöRR?":PRINT"‾EN I öST ELLER VäST?":INPUTA$:GOSUB2160
3430 IFV$="V"ORV$="VäST"THENI=2:IFD%(2)=0THEND%(2)=2:GOTO999
3440 IFV$="ö"ORV$="öST"THENI=7:IFD%(I)=0THEND%(I)=2:GOTO999
3450 IFD%(I)=1THEN4010
3460 IFD%(I)=2THEN4030
3470 GOTO3190
3500 IFD%(D)=0THEND%(D)=2:GOTO999
3510 IFD%(D)=1THEN4010
```

▶

Program "VIRUS"



```
3520 IFD%(D)=2THEN4030
3530 C$="DöRR":GOTO4020
3700 IFPS%<>3THENC$=H$:GOTO4020
3720 IFD%(0)=2THENPRINT"♥KAPET ÄR REDAN öPPET.":GOTO999
3730 IFD%(0)=1THENPRINT"♥KAPET ÄR LÅST.":GOTO999
3740 D%(0)=2:GOTO999
3750 GOTO999
4000 PRINT"■ICK MED NÄSAN RAKT INI EN STÄNGD DöRR.":PRINT"IAR ONT I HUVUDET!!!"
4002 GOTO999
4010 PRINT"■✓AN EJ öPPNA, DöRREN  ÄR LÅST.":GOTO999
4020 PRINT"■♥ER EJ ";C$;" HÄR.":GOTO999
4030 PRINT"■¯öRREN ÄR REDAN öPPEN.":GOTO999
4040 IF H$=""THENPRINT"×AD/VEM SKALL JAG       VÄCKA???":GOTO999
4045 IFPS%<>13THENH$="":GOTO4040
4050 IFH3$="PER"THENS%=1:J%(4,6)=1:GOTO999
4060 IFH3$="KöK"THENS%=1:J%(4,6)=1:GOTO999
4070 PRINT"JAG TROR ATT ";H$:PRINT"REDAN ÄR VAKEN":GOTO999
4080 IF PS%<>13ANDS%=1THENS%=0
4090 RETURN
4100 IFH3$<>"DöR"THENPRINT"■IUR STÄNGER MAN ";H$;".":GOTO999
4110 IFPS%<>7THEN4150
4120 PRINT"×ILKEN DöRR?":PRINT"¯EN öSTRA ELLER VÄSTRA":INPUTA$:GOSUB2160
4130 IFV$="V"ORV$="VÄST"THEND=2:GOTO4150
4140 IFV$="ö"ORV$="öST"THEND=7:GOTO4150
4145 GOTO998
4150 IFD%(D)=0ORD%(D)=1THENPRINT"■¯öRREN ÄR REDAN STÄNGD":GOTO999
4160 D%(D)=0:GOTO999
4200 PRINT"▓▓¯||♠ |♠_ ♠| \¯ \\| "
4205 IFU%=0THENPRINT"▓ NGENTING.":GOTO999
4208 IFI%(23)=0THENPRINT"■NYCKELKNIPPA"
4210 FORI=0TO17
4220 B=I:IFI%(I)=0THENGOSUB2280:PRINT"■";I$
4230 NEXTI
4240 FORI=18TO22
4250 B=I:IFI%(I)=0THENGOSUB2280:PRINT"■¬N ";A3$;A1$;I$
4260 NEXTI
4270 FORI=1TO5
4280 IFJ%(I,1)<>2ANDJ%(I,2)=0THENPRINT"■¬ATIENT:";I
4290 NEXTI
4310 GOTO999
4350 IFH$="PROV"THEN7200
4352 IFU%=5THENPRINT"┌RKAR INTE BÄRA MER!!":GOTO999
4353 PRINT"▓♥LIT&♥LAP"
4355 IFH$=""THENPRINT"×AD SKALL JAG TA?":GOTO999
4360 IFH3$="PAT"ORVAL(H$)<>0THEN4450
4365 RESTORE
4370 FORI=0TO23
4380 READI$
4390 IFH3$=LEFT$(I$,3)THENRESTORE:GOTO4420
4400 NEXTI:RESTORE
4410 PRINT"■✓AN EJ TA ";H$:GOTO999
4420 IFI%(I)=PS%THENI%(I)=0:U%=U%+1:GOTO999
4430 C$=I$:GOTO4020
4440 IFVAL(H$)>5ORVAL(H$=0THENC$="PATIENT "+H$:GOTO4020
4450 IFH3$="PAT"THENPRINT"■×ILKEN PATIENT????":GOTO999
4455 IFVAL(H$)>5ORVAL(H$)=0THENC$="PATIENT "+H$:GOTO4020
4460 IFP%<>0THENPRINT"■┌RKAR INTE SKJUTSA MERÄN EN PATIENT.":GOTO999
4470 IFJ%(VAL(H$),2)=PS%ANDJ%(VAL(H$),1)<>2THENP%=1:U%=U%+1:J%(VAL(H$),2)=0:GOTO
999
4480 C$="PATIENT "+H$:GOTO4020
4600 IFH$=""THENPRINT"×AD SKALL JAG SLÄPPA?":GOTO999
4605 IFU%=0THENPRINT"IAR INGET MED MIG?!":GOTO999
4607 IFH3$="TAN"THEN6600
4610 PRINT"■¬,♥!&__+♥!!!"
4620 IFVAL(H$)<>0ORH3$="PAT"THEN4740
4630 RESTORE
4640 FORI=0TO23
4650 READI$
4660 IFLEFT$(I$,3)=H3$THENRESTORE:GOTO4690
4670 NEXTI:RESTORE
4680 H$="SLÄPPER "+H$:GOTO998
4690 IFPS%=4THEN4720
4700 IFI%(I)<>0THENPRINT"IAR EJ ";H$:GOTO999
4710 I%(I)=PS%:U%=U%-1:GOTO999
4720 IFI%(I)<>0THENPRINT"HAR EJ ";H$:GOTO999
4730 I%(I)=4:U%=U%-1:GOTO999
4740 IFH3$="PAT"THEN4770
4750 IFVAL(H$)>5ORVAL(H$)=0THENPRINT"♥ER EJ PATIENT ";VAL(H$):GOTO999
4755 IFJ%(VAL(H$),2)<>0THENPRINT"■IAR EJ PATIENT ";VAL(H$):GOTO999
4760 J%(VAL(H$),2)=PS%:P%=0:U%=U%-1:GOTO999
4770 FORI=1TO5
4780 IFJ%(I,2)=0THEN4795
4790 NEXTI
4792 PRINT"IAR EJ PATIENT ";VAL(H$):GOTO999
4795 J%(I,2)=PS%:P%=0:U%=U%-1:GOTO999
4800 IFH$=""THENPRINT"■VA!???"GOTO999
4810 IFH3$<>"DöR"THENGOTO998
4815 IFI%(23)<>0THENPRINT"■IAR INGA NYCKLAR!":GOTO999
4830 IFD%(D)=1THEND%(D)=2:GOTO999
4835 IFD%(D)=0THEND%(D)=1:GOTO999
4840 IFD%(D)=2THENPRINT"■¯öRREN ÄR öPPEN!":GOTO999
4900 PRINT"■VILKEN DöRR?":PRINT"¯EN öSTRA ELLER VÄSTRA?":INPUTA$:GOSUB2160
4910 IFV$="V"ORV3$="VÄS"THEND=2:GOTO4815
4920 IFV$="ö"ORV$="öST"THEND=7:GOTO4815
4930 GOTO998
5000 IFI%(23)<>0THEN4815
5005 IFPS%<>3THENC$=H$:GOTO4020
5010 IFD%(0)=0THEND%(0)=1:GOTO999
5020 IFD%(0)=2THENPRINT"■♥KÅPET ÄR öPPET!":GOTO999
5030 IFD%(0)=1THEND%(0)=0:GOTO999
5040 GOTO998
5050 IFH$=""THENPRINT"■×AD SKALL JAG LAGA?":GOTO999
```

▶

```
5060 IFH3$="MAT"THENPRINT"■/öKSPERSONALENS FACK- FöRENING FöRBJUDER MIGATT GöRA
5070 IFH3$="MAT"THEN999
5090 GOTO998
5100 IFPS%<>3THENC$="TELEFON":GOTO4020
5110 IFVAL(H$)=0THEN998
5111 POKE36878,15:FORL=1TO3:FORM=1TO50:POKE36876,230:FORN=1TO5:NEXTN:POKE36876,0
5112 NEXTM:FORM=1TO1000:NEXTM:NEXTL:POKE36878,0
5120 IFH$="154086"THENPRINT"■OM HÄLSAR OCH SÄGER  ATT JAG SKALL ÄTA MIN GRöT PÅ
 MORGONEN
5125 IFH$="154086"THEN999
5130 IFH$="146080"THENPRINT"■HÄLSAR ATT VI SYNS PÅ ¬OLOSSEUM KL SJU":GOTO999
5140 IFH$="420000"THENPRINT"■!ARA EN MASSA DJUR-   LÄTEN I LUREN":GOTO999
5150 IFH$="112213"THENPRINT"■/AGON SKREK TILL MIG  ATT JAG ALDRIG SKULLE RINGA H
ONOM IGEN!
5155 IFH$="112213"THEN999
5160 IFH$="203410"THENPRINT"■LRU /RUT UNDRAR VAD   JAG VILL":INPUTA$:PRINT"■HON
LA PÅ!":GOTO999
5180 IFH$<>"180246"THEN5200
5185 PRINT"■\AN UNDRAR VAD JAG     VILL BESTÄLLA?":INPUTA$:GOSUB2160
5187 IFV3$="XEP"ORH3$="XEP"THENI%(22)=4:GOTO5198
5190 PRINT"■-INNS EJ PÅ APOTEK!!":GOTO999
5198 PRINT"■LEVERERAT VÅRT LAGER":GOTO999
5200 PRINT"■\NGET SVAR!":PRINT"♥KALL JAG FöRSöKA IGEN":GOTO999
5300 IFH$=""THENPRINT"■×AD SKALL JAG LÄSA???":GOTO999
5305 IFH3$="PAP"THEN5400
5310 IFH3$<>"BOK"THEN5500
5315 IFI%(8)<>0THEN5330
5320 GOTO1330SI%=0:GOTO999
5330 PRINT"■ IAR INGEN BOK!":GOTO999
5400 GOTO6200
5500 IFVAL(H$)=0THEN5600
5510 IFI%(8)<>0THEN5330
5520 IFVAL(H$)>12THENPRINT"■ !ITTAR INTE SÅ MÅNGA  SIDOR":GOTO999
5530 SI%=VAL(H$)
5540 IFSI%>12THENH$="100":GOTO5520
5545 ONSI%GOTO1500,1600,1650,1710,1750,1780,1800,1830,1870,1900,1930,2000
5550 GOTO1330
5560 GOTO999
5600 IFH3$<>"SID"THEN5700
5610 IFI%(8)<>0THEN5330
5620 SI%=SI%+1:GOTO5540
5700 IFH3$<>"ADR"THENPRINT"■KAN EJ LÄSA■";H$:GOTO999
5705 IFI%(17)<>0THENPRINT"■ IAR INGEN ADRESSBOK!":GOTO999
5710 GOTO1250
5720 IFI%(8)=0THENSI%=SI%+1:GOTO5540
5730 GOTO998
5800 PRINT"■SÄGA VAD TILL VEM???":GOTO999
5900 IFH$=""THENPRINT"■×AD SKALL JAG TRYCKA  PÅ????":GOTO999
5910 IFPS%<>17THENC$=H$:GOTO4020
5920 IF H3$="KNA"THEN 6000
5930 IFH3$="VAR"ORH3$="KAL"THENPRINT"■×ILKEN ÄR VILKEN??":GOTO999
6000 PRINT"■VILKEN KNAPP?■
6010 INPUTA$:GOSUB2160
6020 IFV3$="RöD"ORH3$="RöD"THENK%=1:GOTO999
6030 IFV3$="BLÅ"ORH3$="BLÅ"THENK%=2:GOTO999
6040 A$="TRYCKER PÅ "+A$:GOTO998
6050 IFJ%(3,2)=17THENJ%(3,5)=1:GOTO999
6100 IFK%=0THENRETURN
6105 IFJ%(3,2)=17ANDK%=2THENJ%(3,5)=1
6110 IFPS%>14ANDPS%<17THENK%=0:RETURN
6120 RETURN
6200 IFI%(7)<>0THENPRINT"■ IAR INGET PAPPER?":GOTO999
6210 PRINT"■⊃♥LARVIGA ANTECKNINGAR":PRINT:PRINT
6220 PRINT"FUNNIT ASXXZ NY VIXXRUS SOM EJ ÄR KÄNNT."
6230 PRINT:PRINT"\|-/¬|  ♥\||♥♠\|!!!!!!!!!!!!":PRINT:PRINT
6240 PRINT"♥ER UTXXXXXXX D RTE %%%% X%":PRINT:PRINT
6250 PRINT"BOTAS XCFDXSVX MED ♣¯T|♠L..CDE FEE
6260 GOTO500
6300 IFH$=""THENPRINT"×AD SKALL JAG TVÄTTA?":GOTO999
6310 IFH3$<>"HÄN"THENPRINT"VAD????? BYXORNA???":GOTO999
6320 IFPS%=1ORPS%=19THEN6330
6325 PRINT"■VAR???":GOTO999
6330 IFSC%<5THENSM%=0
6340 GOTO999
6400 IFSM%<>0THENSC%=SC%+1
6410 IFSM%=0THENSC%=0
6420 IFSC%=20THENPRINT"■¬U HAR BLIVIT SMITTAD AV SMITTSAMT VIRUS!":GOTO3170
6430 RETURN
6500 IFH$=""THENPRINT"■♠LLA HÄR ÄR FöR SJUKA FöR SEX!":GOTO999
6510 IFH3$="LAM"THENPRINT"■¯E ÄR REDAN TÄNDA.":GOTO999
6520 IFH3$="BUN"ANDPS%=1ANDI%(5)=0THENB%=1:GOTO999
6530 IFH3$="BUN"ANDI%(5)<>0ANDPS%=1THENPRINT"■ IAR INGET ATT TÄNDA   MED.":GOTO99
9
6540 IFH3$="TÄN"ANDI%(5)=0THENT%=1:GOTO999
6550 PRINT"■×AD SKALL JAG TÄNDA??":GOTO999
6560 IFT%<>0THENTC%=TC%+1
6570 IFTC%=5THENPRINT"■ \AG BRÄNDE FINGRARNA  SVART./AN EJ FORTSÄTTA ARBETA.":GOT
03170
6580 RETURN
6600 IFI%(5)<>0THENPRINT"VA!!":GOTO999
6605 T%=0:TC%=0:PRINT"OJ!":PRINT"JAG TAPPADE NÅGOT":I%(5)=PS%:U%=U%-1:GOTO999
6700 IF(LEFT$(H$,7)="NYCKELS"ANDPS%=3)ANDI%(23)<>0THENPRINT"¬N NYCKELKNIPPA":GOT
0999
6705 IFH3$="PAT"ORVAL(H$)<>0THEN6980
6710 IF(H3$="JOU"ANDPS%=3)ANDD%(0)=2THENPRINT"¯ET ÄR TOMT OCH DAMMIGT":GOTO999
6715 IFH3$="SPR"ANDI%(0)=0THEN10200
6720 IFH3$<>"VAT"THEN6790
6730 IFPS%<>17THEN6790
6740 PRINT"DET ÄR ";
6750 IFK%=0THENPRINT"INGET"
6760 IFK%=1THENPRINT"VARMT"
6770 IFK%=2THENPRINT"KALLT"
```

Programmet fortsätter på sidan 40

FRÅGA THOMAS

Det första brevet att besvara kommer från signaturen "Undrande" och lyder som följer:

HEJ! Jag är en lycklig ägare av en VIC 64 och har ett par frågor.

1. Hur bär man sig åt för att konstruera en egen teckengenerator i en annan videosektion än den normala, t ex videosektion ett?

2. Vad krävs i hårdvara för att kunna programmera i Assembler?

3. Hur bär man sig åt för att få någonting joystick-manövrerat?

SVAR:

Jag utgår från att du redan vet hur man konstruerar en teckengenerator och vad du vill ha reda på är hur man ändrar videosektion. En teckengenerator upptar 4096 bytes (256 × 8 × 2 = 4096), och en standard sådan finns placerad på adress 53248 och framåt i ROM.

Videochippet kan bara hämta information som ligger inom en och samma videsektion (16k-område). Vilken sektion som ska användas bestäms av de två minst signifikanta bitarna ur adressen 56575 enligt nedan:

Block	Bitvärde	Ramarea
0	11	0-16383 (normalt)
1	10	16384-32767
2	01	32768-49151
3	00	49152-65535

Men det är fler adresser som måste ändras om man vill byta videosektion. Pekare till skärm och teckengenerator samt skärmsida för skärmeditorn. För att få fram adressen till bildminnet tar man värdet av bitarna 4–7, från adressen 53272, multiplicerar detta med 1024 och lägger till videosektionens startadress. Detta värde dividerar man med 256 och lagrar detta i adressen 648.

Detta görs nu om, fast med bitarna 0–3 och utan någon 648-adress, för att placera teckengeneratorn inom videosektionen. Nedan visas ett programexempel för att använda videosektion 1 med skärmen placerad på adress 17408 och teckengeneratorn på adress 18432.

```
10 POKE 56576 ,(PEEK(56576)
AND 252) OR 2
20 POKE 53272 , 19
30 POKE 648, 72
```

Över till nästa fråga angående assemblerprogrammering. Svaret på din fråga är NEJ man behöver inte en speciell hårdvara, men däremot ett program som kodar om assemblerkod till maskinkod. Sådana program ligger i prisklassen 300 kr och uppåt. Att använda joystick som manöverkontroll i ett program är inte speciellt svårt på VIC 64, eftersom uppläggningen av joystickregistren är logiska. Man använder sig av adresserna 56321 och 56320 för att läsa av joystick 1 respektive 2. För att kunna läsa av joystick 2 måste man ställa om ett datariktninsregister (56320) först, och sedan inte glömma bort att ställa tillbaka det i normalläge. Nedan visas ett enkelt avläsningsprogram som skriver ut joystickvärden.

```
10 PRINT 255-PEEK
(50321);TAB (25)
20 POKE 56320,0:PRINT 255-
PEEK (56320):POKE
56320,255
30 GOTO 10
```

Nästa brev som ska besvaras kommer från Joel Grönning och lyder så här:

HEJ! Jag är en datorintresserad kille på 16 år som gärna skulle vilja ha svar på hur man gör interruptrutiner på VIC 64?

SVAR:

Med interruptrutiner menas en maskinkodsrutin som anropas och exekveras 60 gånger i sekunden under tiden att datorn uträttar något annat. Normalt skall datorn anropa en maskinkodsrutin på adressen $EA31. Denna adress ligger lagrad i en annan adress, nämligen $0314. Adressen $0314 kallas för en vektor, med det menas att man hoppar till en startadress som ligger lagrad i denna adressen, vektorn. När man vill lägga in egna maskinkodsrutiner i interruptprocessen så ändrar man hoppadressen i vektorn $0314 till att peka på sin egen maskinkodsrutin. I slutet av denna rutin måste man hoppa till den ordinarie M/C-rutinen på EA31, JMP$EA31. Interruptprocessorn kan liknas vid figur A och B nedan.

Rutinen du vill ska exekveras får inte vara för lång eftersom den då kommer att försinka datorn i dess arbete. Och rutinen måste alltid hoppa till $EA31, annars avstannar datorn. Med lite extra experimenterande kommer du snart att behärska rutinen.

Thomas Wernersson

FIG A

NORMALT

$0314 ——→ $EA31

FIG B

EGEN RUTIN EXEKVERAS

$0314 ↘ $4000 (Egen rutin) ↗ $EA31

TIPS & FÖRSLAG

PROGRAMMERINGSHJÄLP!

```
1 REM*****PROGRAMERINGSHJALPMEDEL*****
2 REM*********************************
3 REM*84.09.10. *Y.L. DATA*          *
4 REM*          *SKALLINGE*          *
5 REM*          ***********          *
6 REM*   PL.122 430 17 SKALLINGE     *
7 REM*   TEL: 0340/35444             *
9 REM*********************************
10 X=36:Z=32
20 K=PEEK(56)*256+PEEK(55)-227
30 V=INT(K/256)
40 FORI=KTOK+226:READP:POKEI,P:NEXT
50 POKE56,V:POKE55,K-V*256:POKE51,K-V*256:POKE52,V
60 H=INT((K+50)/256):L=(K+50)-H*256:POKEK+Z,L:POKEK+X,H
70 FORI=673TO767:READS:POKEI,S:NEXTI
80 PRINT"SYS";K+23"FOR START
90 PRINT"SYS";K"FOR NORMALT"
100 PRINT"TRYCK L OCH SEDAN     RETURN FOR LISTNING  CRSR LISTAR NEDAT"
110 PRINT" OCH SHIFT CRSR UPP.   GA UR MED RETURN     STARTA SEDAN MED L"
120 PRINT" OCH RETURN SA BORJAR  LISTN DAR DU GICK UR"
130 PRINT" DU KAN AVEN ANGE      RAD NR.EFTER L EX:L20 FOR ONSKAD RAD"
140 NEW
150 DATA 169,201,133,124,169,58,133,125,169,176,133,126,169,26,141,6,3,169
160 DATA 199,141,7,3,96,169,0,133,0,169,76,133,124,169,79,133,125,169,29,133,126
,169
170 DATA 202,141,6,3,169,2,141,7,3,96,201,76,208,8,72,165,122,201,0,240,9
180 DATA 104,201,58,144,1,96,76,128,0,32,115,0,240,14,176,38,32,107,201,24
190 DATA 144,35,32,215,202,32,42,197,169,255,197,0,240,9,169,0,133,20,133
200 DATA 21,24,144,8,165,01,133,20,165,02,133,21,24,144,221,76,8,207,32
210 DATA 19,198,160,2,177,95,133,20,200,177,95,133,21,160,0,177,95,201,0
220 DATA 208,11,200,177,95,201,0,208,4,240,190,234,234,32,201,198,32,228
230 DATA 255,201,0,240,249,201,145,240,22,201,80,201,13,240,10,230,20,208
240 DATA 199,230,21,208,195,240,170,160,2,32,242,2,24,144,150,169,0,197,20
250 DATA 208,6,197,21,240,140,198,21,198,20,32,19,198,160,2,177,95,197,20
260 DATA 208,231,200,177,95,197,21,208,222,32,95,229,24,144,161
270 DATA 164,73,41,127,32,9,225,201,34,208,6,165,15,73,255,133,15,200,240
280 DATA 17,177,95,208,17,168,177,95,170,200,177,95,134,95,133,96,96,96,96
290 DATA 108,6,3,16,217,201,255,240,213,36,15,48,209,56,233,127,170,132,73
300 DATA 160,255,202,240,8,200,185,158,192,16,250,48,245,200,185,158,192
310 DATA 48,180,32,71,203,208,245,177,95,133,01,200,177,95,133,02,169,255,133,0,
96
READY.
```

Detta program är ett programmerings-hjälpmedel. Med denna rutin inlagd i din VIC 20 går det att lista programmet du arbetar med på ett lättare och smidigare sätt än vanligt. Åtminstone har jag själv blivit irriterad många gånger på att inte kunna stanna listningen och sedan fortsätta antingen upp eller ner i basicraderna, speciellt om man inte har någon skrivare.

När programmet laddats in och körs kommer en bild upp på skärmen, som visar vilka SYS-adresser som gäller för in- och urkoppling av rutinen (skriv upp dessa). Ladda sedan in det program som du skall arbeta med (om du inte skall börja på ett nytt). Programmet, som är skrivet i maskinkod, kan användas oavsett hur mycket minne du har anslutit till din VIC.

Efter att rutinen är aktiverad med SYS-kommandot, startar du listningen med "L" som i listning och därefter "RETURN". Listningen startar då från första programraden. Vill du börja lista från en speciell rad anger du bara radnumret efter "L" ex:"L 50 RETURN". Fortsatt listning sker när 'CRSR-vertikal" hålls nedtryckt. För listning uppåt (baklänges) tryck "SHIFT" och samma CRSR-tangent. Skärmen kommer då att rensas och raderna listas överst (på skärmen). När listningen backats till önskad rad, fyller du skärmen genom att släppa SHIFT-tangenten något före CRSR-tangenten.

När någon ändring skall göras i programmet, som du arbetar med, tryckt endast "RETURN" och ändra sedan som vanligt. Återgå till listningen med "L och RETURN" så fortsätter listningen där den slutade.

Forts på sidan 33

Kassettmotorstyrning genom POKE-kommandon

I register 1 hos 6510-processorn (outputregistret) sker styrningen. Bit-funktionerna är följande:

Bit 0: omkopplare för Basic-RAM;
Bit 1: omkopplare för driftsystemrutiner;
Bit 2: omkopplare för teckengenerator-ROMs;
Bit 3: omkopplare för datautgångarna;
Bit 4: kontroll av PLAY-tangenten på kassettspelare;
Bit 5: omkopplare för kassettmotorn;
Bit 6+7: ej fastalgda.

Genom A=PEEK (17 el 32:POKE 192A:POKE1,A kopplas motorn ur.
Genom B=PEEK(1)ANDNOT32:POKE192,B:POKE1,B kopplas motorn på under förutsättning att PLAY-tangenten var nedtryckt.
Register 192 innehåller motorns brytarställning och lägesförändring Genom:

```
10 C=PEEK(1)AND 16 och genom fråga
20 IFC=0 THENPRINT" PLAY TANGENT NEDTRYCKT":END
30 PRINT" PLAY TANGENT EJ NEDTRYCKT"
```

Härigenom kan programförloppet göras avhängigt av tangentställningen.

Lägg in diskett-rutiner i dina program! – VIC 64

Ofta kan det vara praktiskt att lägga in en diskett-rutin i sitt basic-program, med möjlighet att läsa directory, radera filer eller formattera nya diskar, – utan att bryta sitt huvudprogram. Läser man in directory med LOAD"$",8 så spolieras ju dessutom basic-programmet, och måste laddas in på nytt.

Här följer ett program med "diskett-rutiner", som förutom ovanstående även läser av ev error-meddelanden från diskettenheten. Man kan lätt länka ihop denna rutin med andra basic-program, (förutsatt att man har några hundra bytes kvar i minnet – och det har man ju som regel).

Följ nedanstående tågordning, så kan du lätt länka ihop diskett-rutinerna med andra program:

1) Tryck in "diskett-rutiner" enligt bifogad programlista. Provkör så att du vet att rutinen fungerar. Spara programmet med namnet "diskett-rutiner" på en disk.

2) Ladda in ditt huvudprogram i datorn (d v s det program du vill förse med diskett-rutinerna).

3) Skriv följande i direktmod:

S=PEEK(45)+PEEK(46)*256-2:H=INT(S/256):L=S-H*256:POKE43,L:POKE44,H

(d v s du ställer om pekaren för basic-start till det första programmets slut)

Lägg disken med "diskett-rutiner" i driven och skriv:

LOAD"DISKETT-RUTINER",8

Återställ pekare för basic-start till normalt värde med:

POKE43,1:POKE44,8

4) Lägg in lämpligt tangent-val från ditt huvudprogram, för att komma till diskett-rutinerna (...GOTO 6000), och ändra återhoppet från diskett-rutinen (rad 6012), till lämplig rad i din huvudmeny.

OBS. a) Att om ditt huvudprogram har rad-nr över 6000 måste diskett-rutinen numreras upp så att den ligger helt ovanför huvudprogrammet. Var i så fall extra noggrann med att ändra goto-gosub-adresserna på korrekt sätt!

b) Om du redan tidigare har "trixat" med startadressen för ditt huvudprogram, så inser du lätt att du även ska återställa pekaren (efter laddningen) på motsvarande sätt. □

BG

TIPS & FÖRSLAG

```
6000 PRINT"♥":PRINT:PRINT:PRINTTAB(7)"* * DISKETT-RUTINER * *":PRINT
6002 PRINT:PRINT:PRINTTAB(7)"F1  SKRIV ERROR-MEDDELANDE"
6003 PRINT:PRINTTAB(7)"F3  RADERA FIL"
6004 PRINT:PRINTTAB(7)"F5  FORMATTERA DISK"
6005 PRINT:PRINTTAB(7)"F7  SKRIV DIRECTORY":PRINT
6006 PRINT:PRINTTAB(7)"<RETURN> ATER TILL MENY"
6009 REM VAL FRAN MENY
6010 GETA$:IFA$=""THENS010
6012 IFA$=CHR$(13)THEN6010:REM ANDRAS TILL RAD-NR FÖR HUVUDMENY
6014 A=ASC(A$)-132:IFA<1ORA>4THEN6010
6015 OPEN15,8,15
6016 ON A GOSUB 6020,6030,6040,6060
6018 CLOSE15:GOTO6000
6019 REM ERROR-MEDD
6020 INPUT#15,A,B$,C,D
6022 IFA>19THENPRINTA,B$,C;D:INPUT"TRYCK <RETURN>";R$
6024 RETURN
6029 REM SCRATCH
6030 PRINT"VILKEN FIL SKA RADERAS";:INPUTY$
6032 PRINT#15,"S0:"Y$:RETURN
6039 REM FORMATTERA
6040 PRINT"ANGE DISK-NAMN";:Y$="":INPUTY$:IFY$=""THENRETURN
6042 PRINT"ANGE ID-KOD";:INPUTID$:IFLEN(ID$)>2THENPRINT"MAX 2 TECKEN!":GOTO6042
G044 PRINT#15,"N0:"Y$","ID$:RETURN
6059 REM DIRECTORY
6060 OPEN8,8,0,"$0":GET#8,A$,B$
6061 PRINT"♥":PRINTTAB(8)"* DISKETT-BIBLIOTEK *":PRINT
6062 GET#8,A$,B$,A$,B$:C=0
6065 IFA$<>""THENC=ASC(A$)
6067 IFB$<>""THENC=C+ASC(B$)*256
6070 PRINTTAB(8)"▓"MID$(STR$(C),2);TAB(12);"■";:REM REVERSE ON RESP OFF
6072 GET#8,B$:IFST<>0THEN6090
6074 IFB$<>CHR$(34)THEN6072
6076 GET#8,B$:IFB$<>CHR$(34)THENPRINTB$;:GOTO6076
6078 GET#8,B$:IFB$=CHR$(32)THENS078
6080 PRINTTAB(30);:C$=""
6082 C$=C$+B$:GET#8,B$:IFB$<>""THEN6082
6084 PRINT"▓"LEFT$(C$,3):IFST=0THEN6062:REM REVERSE ON
6090 PRINT"BLOCK LEDIGA":CLOSE8
6092 PRINT:PRINT"TRYCK <RETURN>";:INPUTR$:RETURN
READY.
```

Forts från sidan 31

"PROGRAMMERINGS-HJÄLP"

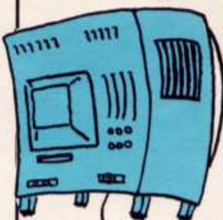
KAN DRÖJA

Om programlistningarna innehåller radnummer med stora hopp ex: 100 och nästa rad heter 6000, kommer det att dröja något när du listar bakåt över detta hopp. När du är klar med ditt program och/eller skall spara det på band eller disk, bör listningsrutinen kopplas ur med den andra SYS-adressen, som angavs vid inladdningen av rutinen. Denna ligger dock kvar och kan aktiveras igen, om du så önskar (och om inte datorn stängs av).

Rutinen upptar endast 225 byte i toppen på RAM + 94 av de 96 lediga byten i minnesområdet 673–767. Detta gör, att om du vill lista ett program med denna rutinen, som använder samma minnesområde för någon subrutin i maskinkod, (som programmet för rättning av stryktips), så kan du inte köra det programmet och sedan försöka lista det med hjälp av denna rutinen, eftersom listningsrutinen då delvis blir överskriven. Vad som händer, om du försöker är, att bara radnummerna listas. Det är dock mycket sällan som denna konflikt uppstår, och om den gör det, så är det inte svårt att kringgå problemet när man vet orsaken. Om du skriver END i stället för NEW på rad 140, så kan du testa, att programmet fungerar, genom att låta det lista sig själv så att säga.

PS. Spara programmet innan du kör, för det är så lätt gjort att skriva fel i datasatserna. DS. □

Y Lindblad



SPELREKORD

Nu sprudlar det av nya spelprogram på marknaden bl a har BALLY/MIDWAY släppt sina spelhalssuccéer "TAPPER" och "SPY HUNTER", men också överraskningen "UP'n DOWN" som nu provar lyckan i programtopplistan (femte-plats) för 64:an!

I nästa nummer har undertecknad utförliga recensioner av dessa "grymma" program! "JUMPMAN" rasar ett steg och snart utses nog LUDWIG till "SWEDEN JUMPMAN MASTER"! Men "ZAXXON" förblir det populäraste programmet för 64:an, det fullkomligt sprutar in rekord! Hur mår lillebror (VIC 20) då? JO då! Även om det inte öser in rekord så försöker nu "AMOK" få plats på softwarelistan!

Ni får gärna skriva eller kopiera av "rutan" (se nedan!), så slipper ni klippa sönder VIC rapport.

SPELREDAKTÖREN C E J





VIC 20

1. AVENGER VIC-1901
1. 27 140 Joakim Jerndal, Skärholmen
2. 10 800 Ken Månsson, Sölvesborg
3. 6 850 Magnus Tibell, Göteborg

2. JELLY MONSTERS VIC-1905
1. 2 601 160 Peter Johansson, Lyckeby
2. 2 499 120 Rudy Birkner, Smygehamn
3. 1 631 860 Pelle Gustavsson, Sundbyberg

3. JUPITER–SUPERLANDER VIC-1907
1. 139 900 Björn Lindman, Huddinge
2. 128 800 Mathias Sånemyr, Stockholm
3. 120 200 Paul Jönsson, Genarp

4. RAT RACE VIC-1909
1. 113 040 Niclas Andersson, Eskilstuna
2. 101 520 Pär-Olof Håkansson, Bjärred
3. 96 200 Olli Pesonen, Angered

5. AMOK (BUG BYTE)
1. 67 350 Jasfd Ragnarsson, Borrby
2. 40 000 Johan Kuuse, Gråbo
3. 26 160 Torleif Tjöre, Västra Lindome

VIC 64

1. ZAXXON (SEGA)
1. 572 780 Martin Malmström, Kävlinge
2. 463 370 Rudy Birkner, Smygehamn
3. 326 550 Per-Olof Karlsson, Filipstad

2. CUP-FINAL (Commodore)
1. 14–0 Peder Isaksson, Ösmo
2. 9–0 Leif Lindström, Örebro
3. 8–0 Magnus Linder, Mölndal

3. JUMPMAN (EPYX)
1. 125 400 Ludvig Gran, Luleå
2. 63 650 Peter Blomgren, Handen
3. 62 100 Sverker Edbladh, Landskrona

4. SNOKIE (FUNSOFT)
1. 33 240 Magnus Linder, Mölndal
2. 30 400 Håkan Åkerberg, Vällingby
3. 30 180 Jens Lund, Borrby

UP'n DOWN (BALLY/MIDWAY)
1. 81 720 SPELREDAKTÖREN C E J
2. 23 040 A & A PRODUKTION
3. 1 140 Martin Kreanzmer, Lindome

VIC 20 □ VIC 64 □ Datum
Namn ....
Adress ....
Postnr .... Postadress ....
Spel ....
Erhållna poäng ....
Intygas av målsman eller annan myndig person.





WORMY – VIC 64

Wormy är ett spel som bygger på "masken". (Det har även vissa likheter med det spel som vann tredje pris i VIC rapports tävling.)

Här gäller det att äta grodor. (Grymt men roligt.) När man ätit upp alla grodor i första nivån så ska man gå vidare till nivå två. Detta gör du genom END-skylten, utan att hamna på kroken (instruktioner finns i spelet).

Masken kan styras i fyra riktningar med joystick, eller tangentbord.

Karl Hörnell



```
READY.

0 GOSUB600
1 POKE53281,LE:PRINT"■";:POKE53281,0:GR=0:POKE1564,48+L:POKE53265,27
2 POKE54296,15:POKE54277,190:POKE54278,248:POKE54276,33:FORT=80TO119
3 POKE1024+T,97:POKE54273,T*2:NEXT:FORT=159TO999STEP40:POKE1024+T,97
4 POKE54273,T/4:NEXT:FORT=998TO960STEP-1:POKE1024+T,97
5 FORT=920TO120STEP-40:POKE1024+T,97:POKE54273,(T-900)*2:NEXT
6 PRINT"■LEVEL:";LE;"    MINI-WORMS:":PRINT"SCORE:";SC;TAB(12);"TIME:"
7 PRINT"■■■■■■| |":POKE1980,79:POKE1981,80:POKE1982,81:POKE1564,32
8 FORT=1TOXA:A=INT(RND(1)*36+2)+40*(INT(RND(1)*18+4))
9 POKE1024+A,73:POKE1064+A,74:POKE55296+A,1:POKE55296+40+A,1:NEXT
10 FORT=1TO3:A=INT(RND(1)*37+1)+40*(INT(RND(1)*20+3))
11 POKE1024+A,75:POKE1025+A,76:POKE1064+A,77:POKE1065+A,78:POKE55296+A,13
12 POKE55297+A,13:POKE55296+40+A,5:POKE55296+41+A,5:NEXT
15 GU=161:POKE53283,R(INT(RND(1)*9+1)):KR=68:HU=67:Q=40:POKE55296+GU,14
16 POKE54284,190:POKE54285,248:POKE54283,17:POKE54280,0:TI$="000000"
17 IFM<>0THENPRINT"■";TAB(23);:FORT=1TOM:PRINT"■■■■■";:NEXT:PRINT"■";
20 K=PEEK(PE):POKE1024+GU,KR
21 IFK=B(1)THENQ=-40:HU=67:KR=65
22 IFK=B(2)THENQ=40:HU=67:KR=68
23 IFK=B(3)THENQ=-1:HU=66:KR=69
24 IFK=B(4)THENQ=1:HU=66:KR=64
30 GU=GU+Q:IFPEEK(1024+GU)<>32THEN100
31 POKE1024+GU,HU:POKE55296+GU,14
80 FORT=1TOF:NEXT
81 PRINT"■■■■■■■■■";SC;TAB(17);RIGHT$(TI$,2):IFTI$="000040"THEN101
82 A=A+1:A=A AND 7:POKE54280,A(A)
99 GOTO20
100 K=PEEK(1024+GU):IFK>74THEN120
101 POKE54280,0:POKE54296,15:POKE54276,129:POKE54273,10:POKE1024+GU,90
102 POKE55296+GU,7:FORT=1TO99:NEXT:POKE54273,20:POKE1024+GU,91:FORT=1TO99:NEXT
103 POKE54273,30:POKE1024+GU,92:FORT=1TO99:NEXT:POKE1024+GU,32:FORT=15TO0STEP-.1
104 POKE54296,T:NEXT:POKE54273,30:M=0:FORT=15TO0STEP-.3:POKE54296,T:POKE53282,T
105 POKE53283,T+1:NEXT:POKE53283,0:POKE54273,0:POKE54276,33:POKE54296,10
106 FORT=1TO900:NEXT:L=L-1:IFL>0THEN119
107 FORT=0TO255STEP2:POKE54273,T:POKE54273,RND(1)*30:NEXT:POKE54273,0
108 POKE54296,10:PRINT"■■■■■■■■■■■■■■■■            GAME OVER!":POKE53282,1
109 FORT=1TO3000:NEXT:GOTO500
119 POKE53265,11:POKE53282,1:GOTO1
120 IFK>78THEN130
121 IFK=75THENPOKE1024+GU,32:POKE1025+GU,32:POKE1064+GU,32:POKE1065+GU,32
122 IFK=76THENPOKE1024+GU,32:POKE1023+GU,32:POKE1064+GU,32:POKE1063+GU,32
```



```
123 IFK=77THENPOKE1024+GU,32:POKE1025+GU,32:POKE984+GU,32:POKE985+GU,32
124 IFK=78THENPOKE1024+GU,32:POKE1023+GU,32:POKE984+GU,32:POKE983+GU,32
125 POKE54280,0:POKE54296,15:FORT=0TO255STEP5:POKE54273,T:NEXT:POKE54273,0
126 POKE54296,10:SC=SC+10*LE:GR=GR+1
129 GOTO31
130 IFGR=3ANDK<>97THENA$=TI$:GOTO132
131 GOTO101
132 POKE54280,0:FORT=15TO0STEP-1:POKE54296,T:FORS=4TO20:POKE54273,S:NEXT
133 NEXT:FORT=1TO1000:NEXT:POKE54273,30
134 FORS=1TO40-VAL(RIGHT$(A$,2)):SC=SC+10:PRINT"■■■■■■■■";SC:FORT=15TO0STEP-1
135 POKE54296,T:NEXT:NEXT:POKE54273,0:POKE54296,10:PRINT"■■■■■■■■■■■■■"
136 PRINT"■   NEW LEVEL! YOU GET A MINI-WORM.■■":M=M+1
137 IFM<5THEN145
138 PRINT"   YOU NOW HAVE FIVE MINI-WORMS!!■":PRINT"                    EXTRA LIFE + 1000"
139 M=0:SC=SC+1000:L=L+1:POKE54273,40:FORT=15TO0STEP-.1:POKE54296,T:NEXT
140 POKE54273,60:FORT=15TO0STEP-.1:POKE54296,T:NEXT:POKE54273,80
141 FORT=15TO0STEP-.1:POKE54296,T:NEXT:POKE54273,0:POKE54296,10
145 LE=LE+1:F=F-5:XA=XA+5:FORT=1TO3000:NEXT
146 IFLE=7THENLE=1
147 POKE53265,11:GOTO1
500 PRINT"■";:IFSC<=O(6)THEN520
501 PRINT"■       ENTER YOUR INITIALS, PLAYER.■■":FORT=1TO11:GETA$:NEXT
502 PRINT"                 ■...■■■■";:Q=0:H$=""
503 GETA$:IFA$=""THEN503
504 IFA$=CHR$(20)THENIFQ>0THENQ=Q-1:PRINT"■.■";
505 IFA$=CHR$(13)THEN510
506 IFQ<3THENH$=H$+A$:Q=Q+1:PRINTA$;
509 GOTO503
510 FORT=6TO1STEP-1
511 IFSC<=O(T)THENO(T+1)=SC:O$(T+1)=H$:GOTO515
512 O(T+1)=O(T):O$(T+1)=O$(T)
513 NEXT
515 T=0
520 PRINT"■"
521 PRINT"               ■MEGAWORM HISCORE LIST■■■"
522 FORT=1TO6:PRINTTAB(11);"■";O(T);TAB(23);"■";O$(T);"■■"
523 NEXT:PRINT"■";
524 IFPEEK(197)=64ANDPEEK(56320)<>111THEN524
525 GOSUB620:GOTO1
599 END
600 PRINT"■■■■■■■■":POKE53280,6:POKE53281,6:POKE53272,21:PRINTCHR$(8);
601 PRINT"                 ■              "
602 PRINT"                 ■ WAIT WHILE  "
603 PRINT"                 ■              "
604 PRINT"                 ■ GRAFIX ARE  "
605 PRINT"                 ■              "
606 PRINT"                 ■ SET,PLEASE  "
607 PRINT"                 ■              "
608 POKE998,1:POKE56333,127:POKE1,51:FORT=0TO511:A=PEEK(53248+T)
609 POKE14336+T,A OR (A*2)AND255
610 NEXT:POKE1,55:POKE56333,129
611 FORT=0TO34*8-1:READA:POKE14848+T,A:NEXT
612 R(1)=6:R(2)=2:R(3)=5:R(4)=4:R(5)=9:R(6)=10:R(7)=11:R(8)=12:R(9)=14
615 A(1)=30:A(2)=28:A(3)=25:A(4)=0:A(5)=25:A(6)=23:A(7)=20:A(0)=0
616 FORT=1TO6:READO$(T),O(T):NEXT
620 PRINT"■■";:POKE53280,0:POKE53281,0:POKE53272,30:POKE53270,24:LE=1:XA=20
621 PRINT"  -   -                                  ":F=30:L=3:SC=0:M=0
622 PRINT"  -    ♦                                 "
623 PRINT"  -    ♦ ────♦ ♦─┤ ♦── ♦── ♦  ──┤"
624 PRINT"  -♦── ♦ -   ♦ ♦   ♦ - ♦ - ♦  --"
625 PRINT"  -♦ - ♦ -   ♦ ♦   ♦ - ♦ - ───┐"
626 PRINT"  -♦ - ♦ -   ♦ ♦   ♦ ── -    --"
627 PRINT" ─── ─── ────  ♦   ♦      - ───■■"
628 PRINT"■          BY KARL HORNELL ■1984■■"
629 PRINT"        ■RETURN............START■"
630 PRINT"        ■SPACE......INSTRUCTIONS■"
631 FORT=0TO255:GETA$:IFA$=" "ORA$=CHR$(13)THEN635
632 POKE53283,T:NEXT:GOTO631
635 IFA$=" "THEN650
636 PRINT"■■■■■■■■■■■■■■■■■■■         KEYBOARD / JOYSTICK ?"
637 GETA$:IFA$="K"THEN643
638 IFA$<>"J"THEN637
639 PRINT"□                O                 "
640 PRINT"           PUT ♠ | IN PORT 2"
641 PE=56320:B(1)=126:B(2)=125:B(3)=123:B(4)=119:GOTO649
643 PRINT"□                A                 "
644 PRINT"                 _"
645 PRINT"               C♥X| V"
646 PRINT"                 ┌"
647 PRINT"                 Z"
648 PE=197:B(1)=10:B(2)=12:B(3)=20:B(4)=31
649 FORT=1TO4000:NEXT:POKE53265,11:RETURN
650 PRINT"■"
651 PRINT"■  LET YOUR MEGA WORM CRAWL AROUND THE■"
652 PRINT"   SHARP FISHING HOOKS TO EAT ALL THREE■"
653 PRINT"   BONUS FROGS AND THEN HIT THE 'END'-■"
654 PRINT"   SIGN,WITHIN FORTY SECONDS. ■FOR EACH■"
655 PRINT"   LEVEL PASSED WITHOUT ANY MISSES YOU■"
656 PRINT"   WILL GET A MINI-WORM. FIVE OF THOSE■"
657 PRINT"   GIVE YOU A NEW LIFE.  ■ONE MISS WILL■"
658 PRINT"   CLEAR ALL YOUR MINI-WORMS.■■"
659 PRINT"   ■GOOD LUCK, WORM-MOVER!"
660 GETA$:IFA$=""THEN660
```

Forts på sidan 43



Bok-
RECENSIONER

Easy Interfacing Projects for the VIC 20

På Prentice-Hall har författarlaget Jim Downey, Don Rindsberg och William Isherwood gett ut en intressant bok, för dem som vill använda sin VIC 20 till annat än spel.

Man säger i förordet vad många vet, nämligen att VIC 20 är en i många stycken bättre maskin att koppla ihop med yttre enheter än t ex VIC 64. Man lovar också att lära ut hur man gör sina egna ROM-baserade program och får igång dem utan väntan på långsam laddning från kassettband.

En god vän till recensenten har använt VIC 20 i en så avancerad applikation som det till sjöräddningstjänsten knutna "SAR"-programmet. Det fungerar bra eftersom VIC 20 har en så snabb processor och den av honom använda VIC 20-datorn står påslagen dygnet runt.

En annan bekant berättade att man använt endast inkromet från VIC 20-maskiner för att montera in andra datoriserade styrutrustningar. Han hade också använt en VIC 20 för processreglering – med gott resultat.

TILL NYBÖRJARE

Men åter till boken som ger många goda tips och anvisningar hur man gör och kopplar tillsatser till sin "lilla" VIC 20. Boken, som är skriven på en lättfattlig engelska, är inget uttömmande verk om avancerad maskinvarukonstruktion. Den riktar sig till nybörjaren/amatören och börjar med enkla exempel.

En genomgång av kapitelrubrikerna ger en god bild av vad man får för pengarna som boken kostar. (Den kan veterligt inte köpas någon annanstans än England, men de flesta bokandlare är glada åt att få hjälpa till att skaffa hem ett exemplar).

Första kapitlet heter Computer Fundametals och det är precis vad det handlar om: grundläggande datorbegrepp. Man gör en genomgång av 6502-processorn och vilka signaler som går till och från 6502:an. Man talar om binära bytes och bit-ar, om datorns anatomi och ger en minneskarta. Skrivcykeln och läscykeln dvs de förlopp när processorn lämnar ifrån sig och hämtar in information samt de olika aktuella signalnivåerna presenteras i utförliga diagram. Dataformat och logik-kretsar presenteras utförligt med historiska utblickar över utvecklingen fram till dagens kretsar. De olika logikfunktionerna är väl beskrivna liksom hur de används i specialkretsarna 7400, 7402 och 7404.

Andra kapitlet berättar om hur man använder maskinspråk på VIC 20. De viktigaste sex register som finns i en 6502 presenteras och man kan läsa om hur man använder hexadecimal adressering internt i maskinen. Maskinspråksformatet får utförlig behandling internt och man tar upp SYS-kommandot i BASIC. Ett enkelt programexempel i BASIC visar hur man enkelt kan POKEa in ett hexadecimalt program i minnet. Det bygger på tekniken att man först omvandlar hex-tal till decimaltal och därefter skriver in värdena i en DATA-sats.

Hur man finner lämpliga platser för att lägga in sina maskinspråksrutiner och hur man använder 60 Hz interrup-vektorn får ett par sidor i slutet av kapitlet.

SJÄLV BYGGER

Tredje kapitlet tar upp VIA, dvs 6522 eller Versatile Interface Adapter-chipet. I ett tydligt diagram visas hur det är kopplat till User Port, användarporten. Ett kopplingsschema för en logikprobe som används för att testa logikkretsar och en utförligt beskrivning hur man själv bygger sin prov-apparat får också ett par sidor. De två I/O-portarna samt registret för in/utmatning tillsammans med en minneskarta över hela VIA-chipet beskrivs som inledning till hur man bygger sin egen s k header, som används för att testa tillståndet hos I/O-ledningarna.

Kapitlet fortsätter med några övningar i digital utmatning tillsammans med tillhörande kommandon och kopplingsscheman för en adapter, som medger användning av ett vanligt standard 22/44 experimentkort. Härefter följer några experiment med digital utmatning och avrundas med

behandling av 6522-klockorna. Man visar hur första klockan kan användas som frekvensgenerator och talar om s k one-shot-mode. Det används t ex för att styra en servo-aktiverande funktion som kräver varierad pulsbredd för att styra dess läge.

Den andra klockans något annorlunda arbetssätt beskrivs och man visar på hur skift-registret kan användas. Styrsignal-ledningarna CB1 och CB2 som finns i användarporten presenteras både i en utförlig tabell och i text.

De olika spänningar som tillåts ligga direkt mot VIC 20s olika portar behandlas utförligt liksom hur man styr högspännings-enheter. De olika likströms- och växelströms-belastningar som får användas utan risk för skador på utrustningen behandlas med utgångspunkt från den i England vanliga hushållsspänningens 110 volt. Här måste alltså en svensk användare vara uppmärksam på dimensioneringen av de i kretsschemana inlagda motstånden och transistorerna. Kapitlet avslutas med hur man känner av inmatning både med låga nivåer och biPpolärt, liksom hur man känner av högspända signalnivåer.

TALSYNTES

Fjärde kapitlet behandlar talsyntes. Man presenterar ett billigt tre-chips-system från National Semiconductor, kallat DIGITALKER. Dess funktion är utförligt dokumenterad och den duktige elektronik-hobbyisten bör kunna bygga en egen DIGITALKER med ledning av kopplingsscheman och text. Hur tillsatsen programmeras, ges intelligenta fraser och får utökad vokabulär dokumenteras i både program och flödesschema i slutet av kapitlet.

För lödkolvs-artisten följer sedan några intressanta kapitel om olika mekaniska och elektroniska tillsatser, som kan aktiveras med VIC 20. Hur man styr ett servo, kopplar in det och programmerar det samt hur det kan användas under interrupt-styrning blir detaljrikt omskrivet. Multipla servon, stegmotorer och styrning av dem blir fråntaget varje hemlighet – både programmässigt och mekaniskt.

Forts på sidan 38

COMPUTE
COMPUTE'S Gazette.
No. 1, Januari 1985.

I ärlighetens namn måste man väl medge att det inte hänt så mycket på maskinsidan på hemdatorområdet sedan VIC 64 kom ut för ett par år sedan. På COMPUTE's ledarsida ryktas det nu om häftiga maskiner både från Commodore och Atari. Det skall bli spännande att se vem som kommer först och, framför allt, vad det blir. Det lär i varje fall bli ett strå vassare än Sinclairs QL.

SPELLISTNINGAR

"Paratrooper" är ett spel som går ut på att släppa fallskärmshoppare ner på små öar. Ju mindre ö desto högre poäng om man träffar rätt. Det hela kompliceras av att olika fallskärmshoppare väger olika och att det blåser. Listningar finns för både VIC 20 och 64.

"Rescue of Blondell" är ett spel helt i maskinkod. På sin flygande matta ska man ge sig iväg för att rädda kungens dotter, Blondell, som sitter fången i ett högt torn. Men tornet vaktas av elaka fåglar som beskjuter dig med blixtar. Som försvar kan du själv skjuta blixtar. Guld och Gröna skogar, eller åtminstone höga poäng, till den som lyckas rädda prinsessan. Listning finns till VIC 64.

NYTTA

"Guitar Tuner" – stäm din gitarr med hjälp av datorn. Basiclistning finns till VIC 64.

"Turbo Tape" är enligt COMPUTE! ett av de bästa program de någonsin publicerat. Programmet, som är helt i maskinkod, gör att man kan spara, ladda och verifiera program till bandspelaren **lika snabbt som till disk drive**. Programmet är en guldgruva om du inte har råd att köpa en diskdrive och tycker att bandspelaren är "seg". Decimal maskinkodslistning (Basicladdare) finns för både VIC 64 och VIC 20.

SPEL

Samtliga spel i Gazettens januarinummer finns listade för både VIC 64 och VIC 20.

"Trap 'Em" är ett enkelt, men finurligt, spel för en eller två spelare. Det hela går ut på att bygga stängsel för att på så sätt inhägna sin motståndare. En god portion strategiskt tänkande behövs för att lyckas i det här spelet.

TIDNINGS-RECENSION

"Chomper" är spelet där man styr den lilla roboten (Chomper) med joystick för att samla in energikristaller på en främmande planet full med elaka soldater som står i vägen. Det hela är på tid.

"Kablam" är, som namnet antyder, ett spel det smäller om. Din uppgift är att med hjälp av en joystick fånga de fallande bomberna i en spann med vatten. Lyckas man inte så smäller det! Spelet som är helt i maskinkod kräver snabba reflexer hos spelaren.

"Math Dungeon" är ett exempel på s k "utbildningsspel". I det här fallet gäller det matematik och spelinslaget består i att klara sig undan diverse frågvisa monster. Spelet är av adventuretyp vilket här innebär att man måste kunna lite engelska också för att klara sig.

NYTTA

"Disk Merge" är ett användbart program om du har en disk drive och vill lägga ihop sparade programmoduler till ett program. Rätt använt kan det spara åtskilliga timmars tråkigt inknappningsarbete åt dig. Rekommenderas! □

Forts
"Easy interfacing"

Hur man använder 12 V likström avrundar det innehållsrika kapitel fem. Sjätte kapitlet ägnas åt analog till digital omvandling. Hur ett ADC-chips fungerar beskrivs noga (Analog och Digital Converter, betyder ADC) med exempel från ADC0816-chipet från National Semiconductors.

SCHMITT TRIGGER

Hur man använder en vanlig kassettbandspelare tillsammans med sin VIC 20 beskrivs i kapitel 7. Där får man veta vad man skall leta efter för slags bandspelare, hur man lagrar information på kassettband och vad en Schmitt Trigger är. (För dig som inte känner till denna enkla grej kan berättas att det är en mojäng, som tar bort den rundade formen på signalen från ett kassettband och återger den fyrkantform, som datorn helst vill ha.)

Man får i detta kapitel noggranna anvisningar hur man bygger sitt eget interface (sin mellankoppling alltså) till en vanlig bandspelare. Hur man testar och kör den och hur man letar fel.

Stegvis leds vi i boken upp mot höjderna. Kapitel 8 talar om för oss hur man programmerar EP-ROM-moduler efter att först ha talat om för oss vad som är skillnaden mellan ROM, PROM, EP-ROM och EEPROM.

Här kan man lära sig att programmera maskinvaran, hur den fungerar, hur man bygger och testar sin egen ROM-brännare. Man får veta hur man väljer ut ett lämpligt EPROM och hur man programmerar 2532-chipet från Motorola. De behövliga programmet upptar flera sidor!

Fortsätter man in i kapitlet får vi veta hur man läser av ett ROM, och psst! även hur man omvandlar en specialcartridge till en EP-ROM. Viktigast är hur man skapar sitt eget automatiska startsystem och vi får ett exempel på ett auto-start ROM från BASIC.

Mot slutet får vi lära oss hur man kopplar in en parallell-skrivare, kopplar samman maskinvaran och skriver program för att skriva i Text-mod. Program för grafik-mod presenteras i maskinspråk på fyra sidor!!!

Tionde kapitlet ägnas helt spel-porten. Här får vi lära oss allt om maskinvaran, om joysticks och paddlar. Men vi får också läsa om hur man använder en snabb-koppling, om optiska detektorer och om infraröd-detektorer. Fotoceller och en "fattig-mans-A/D-omvandlare" avslutar kapitlet.

Användning av RS 232-porten på VIC 20 ägnas hela kapitel 11. Baud-rate, nivå-skiftning, handskakning, paritet – allt är utförligt förklarat. Hur man gör ett interface till en apparat vilket som helst samt hur man konstruerar ett nivå-skiftande interface till RS232-porten, hur man använder 20 mA Current Loop förklaras tillsammans med några ord om vilka överväganden som bör ske vid programmeringen. De olika kommandon som kan användas med ett RS232-interface förklaras utförligt, liksom hur man kan kan använda RS232-porten för att ladda in BASIC-program. Kapitlet avslutas med några ord om VIC 20 som en stum terminal.

Nästa och sista kapitlet, nr 12, behandlar VIC 20 som modem-ansluten terminal. Konstruktion av ett enkelt hembyggt modem liksom av en s k akustisk vagga beskrivs med utförliga ritningar. Testning och kommunikationsprogram avslutar genomgången.

Sist i boken finns ett tvåsidigt appendix med alla skärmkoderna för VIC 20. Ett fyrsidigt stickordsregister gör det lätta att hitta tillbaka till de delar där man vill fräscha upp minnet igen.

Boken har ISBN-nr 13-223421-1 och är utgiven i oktober 1984 på Prentice/Hall International. □



Bok-RECENSIONER

LOGO – nybörjarspråk för barn av alla åldrar

Redan 1967 utvecklades den första versionen av programspråket LOGO. Det var prof. Seymour Papert vid Massachusetts Institute of Tecknology som fick idén till detta. Han ville skapa ett språk som gjorde datorn tillgänglig även för barn från 3-årsåldern.

Allt sedan dess har LOGO framför allt förknippats med ett "datorernas barnspråk". Men, som ordspråket säger, "alla barn i början", – och LOGO är en mycket lämplig start även för alla de vuxna som har så stor respekt för datorer att de gärna blockeras av rädsla för att "göra fel".

LOGO är hur enkelt som helst. Lär dig i början 4–5 ord (vanliga engelska ord som DRAW, FORWARD, BACK, LEFT, RIGHT), och du kan roa dig länge. Du får tillgång till en högupplösningsskärm, som du sedan fyller med grafik, allteftersom "paddan" (the turtle) traskar omkring på

skärmen efter dina kommandon. Att programmera motsvarande i Basic är en uppgift för proffs!

Och LOGO tillåter att man gör fel! Den talar helt enkelt om, att t ex den eller den rutinen "needs more inputs", eller att en viss rutin "doesn't like --- as input". Vänligt men bestämt. Inga kraschade program eller hängda datorer här inte!

ÄVEN ARITMETIK

Även om LOGO främst betraktas som ett språk för enkel tillgång till grafik, innehåller det även rutiner för numeriska – resp. strängvariabler samt aritmetiska funktioner (alla vanliga räknesätten samt t ex sinus, cosinus, kvadratrötter). Vidare finns inmatningsrutiner motsvarande Input och Get (= REQUEST (RQ) och READCHARACTER (RC)).

Det finns även rutiner för vissa enkla manipulationer av strängvariabler. Däremot tycks ej LOGO klara av att arbeta med indexerade variabler eller matriser.

För att ytterligare underlätta för användaren kan dessutom flertalet av de vanligaste kommandona förkortas till endast två bokstäver.

Principen för programmering är att användaren definierar egna "ord", som utför något. Dessa definitioner eller procedurer kan enkelt sparas till disk och sedan användas i nya applikationer. (På så sätt finns vissa likheter med Forth – som den uppmärksamme VIC Rapport-läsaren vid det här laget bör vara rätt hemmastadd vid.)

Detta – och mycket mer – kan du läsa i en helt nyutkommen bok "Commodore 64 Logo Primer" av Gary G. Bitter och Nanny Ralph Watson (Reston Publishing Comp, Inc, Virginia, USA).

ELEMENTÄRT

Nära hälften av boken upptas av en MYCKET elementär och detaljerad introduktion till språket, med enkla exempel, ibland nästan alltför triviala. I följande huvudavsnitt som kallas "Quickstart", ges i stället en kort sammanfattning, för den som har tidigare erfarenhet av LOGO.

Ett speciellt avsnitt ägnas en allmän bakgrund till hur programspråket utvecklats och hur det använts och testats bl a i undervisning i USA, – med mycket goda resultat. Och beträffande LOGO som språk för datorn i hemmet anförs hur mycket bättre det är om barnen skapar grafik på en videoskärm än att de passivt sitter framför TV. Detta är speciellt intressant med den ökade medvetenhet om TV-tittandets negativa effekter på barns inlärnings- och föreståelseförmåga.

Boken ger en god introduktion till att använda LOGO, främst för den totala novisen, som därigenom får lättare att närma sig datorer i allmänhet. Samtidigt beskrivs även mer avancerade tekniker som finns tillgängliga i LOGO, vilket kan stimulera även en gammal Basic-räv att titta lite närmare på detta speciella språk.

(Det är säkert inte bara en slump att Christensen & Co valt att lägga in en variant av LOGO's turtlegrafik i Comal, – och kanske har det även bidragit till skolöverstyrelsens val av Comal som skolspråk på datorer i svenska skolor.) □

/BG

Sekventiella filer och Random access

Varje programmerare stöter snart på behovet att lagra data ur ett program till ett sekundärminne – kassettband eller disk. Det är då som data-filer blir aktuella.

Den naturliga metoden när man har begränsade material är att använda SEKVENTIELLA filer – hela datamängden skrivs resp läses varje gång. Datamängden måste givetvis begränsas till det som ryms i datorns primärminne, och om det gäller stora material, kan det vara tidsödande att ladda in en hel registerfil, när enstaka data ska kontrolleras eller ändras.

För den som arbetar med diskettenhet kommer därför snart behovet att skriva och läsa data till/från disketten med random access. Detta innebär att poster av data kan skrivas och hämtas var som helst på disken, utan att man behöver läsa in hela filen.

Förutom snabbare tillgång till enskilda poster, ger random access även möjlighet att lagra mycket större material – ja en hel diskett med drygt 180 K (180 000 tecken) kan användas för ett enda register!

Många som prövat på att arbeta med sekventiella filer och registerprogram, upplever säkert det här med direktfiler eller "random access" som något mystiskt, – något som är till bara för "de stora programmerarna".

En bidragande orsak till detta kan vara den lindrigt uttryckt snåriga famställning som görs i User's Manual till 1541-driven. Snårigheten hänger samman med att man börjar framställningen med "random files" och avslutar med "relativa filer".



Bok-RECENSIONER

UPPRÄTTA SJÄLV

I den första typen, "Random Files", måste programmeraren själv upprätta ett system för att hålla rätt på vilket block data ska skrivas–läsas på, och sedan "notera" detta i BAM – det interna registret på disketten som håller rätt på lediga och upptagna block. Det är

Forts på nästa sida

Forts från sidan 29

Program "VIRUS"



```
6780 GOTO999
6790 IFH3$="HYL"ANDPS%=4THEN6850
6800 IF(H3$="ODL"ORH3$="PRO")ANDPS%=1THEN6910
6810 IFH3$="ADR"ANDI%(17)=PS%THEN6840
6815 IFH3$="PAP"ANDPS%=14THEN6840
6820 IFH3$="BOK"ANDI%(8)=PS%THEN6840
6830 PRINT"SER INGET SPECIELLT":GOTO999
6840 PRINT"KANSKE SKULLE LÄSAS":GOTO999
6850 FORI=18TO22
6860 IFI%(I)=4THENPRINT"┐N BURK MÄRKT ";
6870 IFI%(I)<>4THENPRINT"┐N HYLLA MÄRKT ";
6880 B=I:GOSUB2280:PRINTI$
6890 NEXT:RESTORE:GOTO999
6910 IFI%(4)<>0THEN6830
6920 POKE36879,8:PRINT"▀▌   ╲╱┘┌♥╱┌┐│╲L─":S=4096:C=37888
6930 IFPR%=1000THEN6970
6940 FORI=0TO10
6950 X%=INT(RND(TI)*500):POKES+X%,PR%:POKEC+X%,1
6960 NEXTI
6965 I%(4)=4:U%=U%-1:PR%=1000
6970 GOTO1210
6980 FORI=1TO5
6990 IFJ%(I,2)=0THEN7010
7000 NEXTI:GOTO6830
7010 PRINT"┐♠│╲─╱ +─ ";
7020 IFJ%(I,1)=0THENPRINT"─┼─":GOTO999
7030 IFJ%(I,1)=1THENPRINT"♠LLX♠┴╲││ ♥╲╱"
7040 IFJ%(I,1)=3THENPRINT"─╲♥╱"
7050 PRINT:PRINT"ILODTRYCK:╱┌─╲♠L│":PRINT:PRINT"IEMP:";
7060 IFJ%(I,5)=0THENPRINT"┌NORMALT LÅG"
7070 IFJ%(I,5)=1THENPRINT"╱┌─╲♠L"
7080 IFJ%(I,5)=2THENPRINT"┌NORMALT HÖG"
7090 PRINT:IFJ%(I,6)=0THENPRINT"┐♠│╲─╱ +─ │╱╱─╲│"
7100 GOTO999
7200 IFP%=0THENPRINT"┐♠ MIG SJÄLV?":GOTO999
7210 IFI%(4)<>0ORI%(3)<>0THENPRINT"UTRUSTNING SAKNAS":GOTO999
7220 FORI=1TO5
7230 IFJ%(I,2)=0THEN7250
7240 NEXTI:GOTO998
7250 PR%=J%(I,3)
7260 IFJ%(I,3)=37ANDI%(1)<>0THENSM%=1
7265 I%(3)=4:U%=U%-1
7270 GOTO999
7300 IFH$=""THENPRINT"JAG DRICKER INTE":GOTO999
7305 IFI%(0)<>0THENPRINT"XA???":GOTO999
7310 IFH3$<>"SPR"THENPRINT"VA???":GOTO999
7320 PRINT"VAD SKALL JAG FYLLA   DEN MED?"
7330 INPUTA$
7335 GOSUB2160
7340 IF(V3$="RIB"ORH3$="RIB")ANDI%(18)=0THEN SP%=1
7350 IF(V3$="CLO"ORH3$="CLO")ANDI%(19)=0THENSP%=2
7360 IF(V3$="TRO"ORH3$="TRO")ANDI%(20)=0THENSP%=3
7365 IF(V3$="GIF"ORH$="GIF")ANDI%(2)=0THENSP%=6
7370 IF(V3$="LAR"ORH3$="LAR")ANDI%(21)=0THENSP%=4
7380 IF(V3$="XEP"ORH3$="XEP")ANDI%(22)=0THENSP%=5
7390 IFSP%<>0THEN999
7400 PRINT"VET EJ HUR MAN FYLLER ";A$:GOTO999
7500 IFH3$<>"SPR"THENPRINT"VAD SKALL JAG GÖRA??":GOTO999
7520 IFI%(0)<>0THENPRINT"HAR INGEN SPRUTA!":GOTO999
7530 FORI=1TO5
7540 IFJ%(I,2)=0THEN7580
7550 NEXTI:PRINT"JAG GAV SPRUTAN":PRINT"TILL MIG SJÄLV":PRINT
7555 IFSP%=0THENPRINT"L╱┘ ╲ IL┌┐╲L┌┬┬│":PRINT
7560 PRINT"JAG DÖR.....":PRINT
7570 GOTO3170
7580 IFJ%(I,3)=38ANDSP%=1THEN7660
7590 IFJ%(I,3)=39ANDSP%=2THEN7660
7600 IFJ%(I,3)=0 ANDSP%=3THEN7660
7610 IFJ%(I,3)=35ANDSP%=4THEN7660
7620 IFJ%(I,3)=37ANDSP%=5THEN7660
7625 IFSP%=0THENPRINT"L╱┘ ╲ IL┌┐╲L┌┬┬│":PRINT
7630 SP%=0:I%(0)=4:U%=U%-1:J%(I,1)=0:J%(I,0)=N%
7640 PRINT:PRINT"┐ATIENT DOG":PRINT"AV FELMEDICINERING":GOTO999
7660 SP%=0:I%(0)=4:U%=U%-1:J%(I,1)=3:GOTO999
7700 X%=0:FORI=1TO5
7710 IFJ%(I,1)=0THENX%=X%+1
7720 NEXTI
7730 IFX%<3THEN7770
7740 PRINT;X%;"■ PATIENTER HAR     AVLIDIT:":PRINT
7750 PRINT"♥┌─╲♠L♥│ │─┐♥─╱ │♠─":PRINT"│─♥L╱♠│ ♠││ ":PRINT"♠│ ─╲ +─ L+╲┐╲│":PRINT"♥┌╲ L+╱♠─.":PRINT:PRINT
7760 GOTO3170
7770 RETURN
7800 FORI=1TO5
7810 IFJ%(I,1)<>0THEN7830
7815 J%(I,0)=J%(I,0)+1:IFJ%(I,0)<20THEN7830
7820 IFJ%(I,2)<>18THENPRINT"─┐ L╱│♠─ ═─+┴╲││":GOTO7840
7830 NEXTI
7840 RETURN
8000 FORI=1TO5
8010 IFJ%(I,3)=37THENJ%(I,0)=J%(I,0)+1
8015 IFJ%(I,0)>900THEN8040
8020 NEXTI
8030 RETURN
8040 J%(I,1)=0:RETURN
9000 IF(D%(4)=2ANDPS%<>4)ANDPS%<>10THEN9020
9010 RETURN
9020 IFJ%(4,2)=0OR(J%(4,2)=PS%)THEN9010
9030 IFJ%(4,1)=0THEN9010
```

▶

```
9040 J%(4,1)=0:J%(4,2)=4
9050 RETURN
10000 RN%=0
10010 FORI=1TO5
10020 IFJ%(I,1)=0ORJ%(I,1)=3THENRN%=RN%+1:NEXTI
10030 IFRN%=5THEN10050
10040 RETURN
10050 PRINT"▔ ╱ ♠─ ▕▔▏♠▕":PRINT"▔♠▕╲▔╱▕▔─╱♠.":PRINT
10060 PRINT"▒▕─♠▕▕╲●▕▕▒":PRINT:PRINT
10130 GOTO3170
10200 IFSP%=0THENPRINT"▔EN ÄR TOM.":GOTO999
10210 PRINT"▔EN ÄR FYLLD.":GOTO999
978 CLR
20000 OPEN2,1,1:PRINT#2,D;",";N%;",";PS%;",";S%;",";D0%;",";D1%;",";B%;",";L%;",
";U%
20010 PRINT#2,D;",";U%;",";P%;",";SI%;",";K%;",";PR%;",";SM%;",";T%;",";TC%;",";
SP%
20020 FORI=0TO9:PRINT#2,D%(I):NEXT:FORI=0TO23:PRINT#2,I%(I):NEXT:FORI=1TO5:FORJ=
1TO6
20030 PRINT#2,J%(I,J):NEXTJ:NEXTI
20040 CLOSE2
20050 GOTO999
30000 OPEN2,1,0:INPUT#2,D,N%,PS%,S%,D0%,D1%,B%,L%,U%
30010 INPUT#2,D,U%,P%,SI%,K%,PR%,SM%,T%,TC%,SP%
30020 FORI=0TO9:INPUT#2,D%(I):NEXT:FORI=0TO23:INPUT#2,I%(I):NEXT:FORI=1TO5:FORJ=
1TO6
30030 INPUT#2,J%(I,J):NEXTJ:NEXTI:CLOSE2
30040 GOTO500

READY.
```

□

Forts
"Sekventiella filer..."

som att själv behöva hålla rätt på lediga block, och upprätta "directory" när vanliga program ska sparas. Detta sköts ju (tack och lov) i normala fall av diskettenhetens DOS (Disk Operativ System).

Att använda den andra typen av random-access-filer, "Relativa filer", är betydligt enklare. Här använder man DOS-ens inbyggda system att välja och skriva block, och att markera "upptaget", när ett block på disketten tagits i bruk. Totalt finns 720 pekare, och varje post av data behöver en pekare. Detta innebär att maximalt 720 poster med vardera högst 254 tecken kan användas. Men detta är just precis det som ryms på en hel diskett, varför även relativa filer kan användas för att lagra datamängder som är lika stora som hela disketten (180 K).

Den som vill börja med random access-filer gör klokt i att börja med relativa filer, d v s överlåta det lite mödosamma arbetet med att disponera disketten, till systemets eget DOS – för de flesta ändamål är den mycket bra på detta.

MÅNGA EXEMPEL

En nyutkommen bok av David Miller, "Commodore 64 Data Files – a Basic Tutorial" (Reston Publishing Company, Ink) ger en mycket lättfattlig introduktion till hela detta stora område: datafiler. Boken omfattar ca 400 sidor, men en betydande del av utrymmet åtgår till programexempel, som först presenteras bitvis i löpande text, och sedan sammanfattas i slutet av varje kapitel. Dessa exempel är skrivna mycket spatiöst och med rikliga REM-satser som kommentarer vilket gör dem lätta att läsa.

Bokens första hälft behandlar sekventiella filer, vilket väl de flesta som programmerat kommit i kontakt med. Dessa avsnitt går att tillämpa på såväl kassettband som diskett. Den andra hälften av boken tar upp Random files (relativa filer), och lyckas med ett flertal enkla exempel avdramatisera och förklara sammanhangen.



Författaren skriver i inledningen till detta avsnitt, att första hindret för att förklara random-access-filer är fruktan. Folk är rädda att random access är för svårt för dem att kunna lära sig. Den här boken lyckas verkligen med uppgiften att lotsa läsaren över de mest besvärliga initiala hindren.

Enkelheten beror delvis på att författaren begränsar sig till de "relativa filerna", dvs den andra typen av random-access-filer som bekskrivits ovan.

Trots omfånget, eller kanske just därför, är boken alltså mycket lättillgänglig. Varje problem beskrivs och benas upp – ibland nästan för långt, så att det kan verka banalt. Men alltid torde även dessa "banaliteter" svara på någons frågor.

Här är en kort sammanfattning av att öppna-läsa-skriva till relativa filer:

1. Öppna kommandokanalen till driven:
ex: OPEN 1,8,15 – och sedan,

2. öppna en relativ fil, där max längden av posterna anges (d v s hur många tecken som en post ska kunna innehålla):
ex: OPEN 3,8,3"registerdata,L, "+chr$(max)

3. Att sedan skriva (läsa) posten, görs med angivande av postnummer (PN) i form av lågbyte (LB) och högbyte (HB).
Dessa tal fås på följande sätt:

HB=INT(PN/256)
LB=PN–(256*HB)

Man anger också var det aktuella fältet (den enskilda datan) i posten börjar (PB). D v s hur många tecken från postens start som det ligger. Första fältet har nr 1, och om fält nr 1 innehåller 15 tecken börjar fält nr 2 på på tecken 16, osv. Allt detta anges med ett Print# kommando till kommandokanalen, följt av "P", samt chr$ med datakanal-nr (i detta ex = nr 3), sedan låg- och högbyte, och var i posten (PB):
Ex:
PRINT#1,"P"CHR$(3)CHR$(LB) CHR$(HB)CHR$(PB)

4. Därefter skriver man med:
Ex: PRINT#3,"DATA"

eller läser mud:
Ex: INPUT#3,"DATA"

Efter avslutad skrivning eller läsning avslutas med att stänga filen:

Ex: CLOSE3:CLOSE1

Vad jag har svårt att förstå är en detalj i boken, en vänterutin som väntar på att RETURN-tangenten ska tryckas ned. Med 13 pokekommandon och ett PEEK lyckas författaren krångla till en rutin, som (såvitt jag förstår) inte uträttar annat än följande vänterutin:

```
10000 REM VÄNTA PÅ
<RETURN>
10010 GET A$:IF A$<>
CHR$(13) THEN 10010
10020 RETURN
```

I gengäld fann jag en behändig teknik att utnyttja DOS Wedge (från test-disketten till 1541-driven) – inifrån Basicprogram. I vanliga fall förstörs ju basicprogram, om directory från disken läses in och listas. Men med DOS Wedge inläst och aktiverad kan man läsa directory med
@$<return>
– utan att basicprogrammen förstörs.

Om du bara ser till att ha laddat DOS Wedge och sedan ditt basicprogram, kan du utnyttja denna rutin för att läsa directory inne i basicprogram, med följande programrader:

```
9000 REM LÄS DIRECTORY
9010@"$":PRINT
CHR$(145)CHR$(13)
9020 GET A$: IF A$=""
THEN 9020
9030 RETURN
```

Testa detta i något av dina basicprogram ska du se hur fint rutinen fungerar! Det gör f ö hela denna bok, som varmt kan rekommenderas. □

BG

WORMY

Forts program "~~VIRUS~~"

```
661 GOTO620
699 RETURN
800 DATA20,165,169,169,169,169,165,20,20,85,105,170,170,170,170,40
801 DATA28,28,255,255,255,255,28,28,80,255,255,125,125,255,255,60
802 DATA40,170,170,170,170,105,85,20,20,90,106,106,106,106,90,20
805 DATA255,255,132,158,134,230,134,255,255,255,34,106,98,98,106,255,255,255,33
806 DATA179,51,115,179,255
807 DATA0,24,36,36,24,24,24,12,76,70,198,198,230,124,56,0
808 DATA0,0,0,4,7,15,15,63,0,0,0,16,208,240,240,252,63,95,239,247,251,125,61,15
809 DATA252,250,247,239,223,190,188,240,255,255,195,207,199,207,195,255
810 DATA255,255,51,19,3,35,51,255,255,255,15,55,55,55,15,255
811 DATA24,60,126,24,24,24,24,24,0,32,96,255,255,96,32,0,0,4,6,255,255,6,4,0
812 DATA24,24,24,24,24,126,60,24,24,24,24,255,255,24,24,24
815 DATA0,7,15,15,15,15,15,15,15,15,6,6,127,255,255,127,0,0,0,24,254,255,255
816 DATA254,146,84,56,255,56,84,146,16,133,66,36,152,25,36,66,137,16,16,16,254
817 DATA16,16,16,0
830 DATA1,2,7,0,3,1,3,3,3,1,13,58,103,127,126,57,192,96,240,48,240,224,192,192
831 DATA192,225,241,243,239,222,56,240,204,102,51,153,204,102,51,153
1000 DATA HI,6120,FRE,5470,DRI,4900,KWO,3800,OTZ,2470,"KHH",1100
READY.
```

RÄTTELSE

Angående programmet PROGRAMMERINGSHJÄLP som publicerades i VIC rapport årg 3 nr 11/12. Datasatserna längst ut på rad 160 och 310 saknas.

```
160 DATA 199,141,,7,3,96,109,0,133,0,169,76,133,124,169,79,133,125,169,29,133,126,169

310 DATA 48,180,32,71,203,208,245,177,95,133,01,200,177,95,133,02,169,255,133,0,96
READY
```

Ändring av uppgift i VIC rapport 11/12-84

På sid 78 under rubriken "Visste du att du kan ladda dina VIC 20 program på en VIC 64 eller tvärtom" har ett fel insmugit sig.

Det fanns redan i källmaterialet och kan därför ej lastas på den ökände Tryckfels-Nisse. Det motstånd som skall lödas in före kassettmotorn skall vara på 17 Ohm. Även 15 Ohm fungerar utmärkt.

Ett varmt tack till Johan Björklund, som prövat sig fram. För att lära av Johans misstag, så är det inte likgiltigt om du lödar in motståndet efter kassettmotorn. Det måste vara **före**! □

FAMILJEN VIC TORSSON













VIC rapport

Forts från sidan 9 program "STOPIT"



```
2000 REM ***** PRINT NUMBERS *****
2100 RESTORE:NU=49
2110 FORL=1TO9:FORI=1TO4:READA:POKEA-80,NU:POKEA-80+CS,L:NEXT:NU=NU+1:NEXT
2200 REM PRINT SCOREBOARD 1
2205 POKE53272,21:REM UPPER CASE
2210 IFPG>HITHENHI=PG
2215 CU$="[QQQQQQQQQQQQQQQQQQQQQQQQ]"
2220 PRINTCU$"[S]SCORE:"PG;:PRINTCU$TAB(11)"[R]PLAY - HIT ANY KEY";
2225 PRINTCU$;TAB(30)"[S]HIGH:"HI"[S]"
2300 POKE198,0:WAIT198,1:WAIT197,64
2330 PRINTCU$;TAB(11)"[R]     STOPIT        [S]"
2400 RETURN
2500 DATA1400,1364,1598,1534,1320,1284,1454,1724
2505 DATA1554,1790,1294,1494,1234,1754,1770,1330
2530 DATA1200,1245,1844,1880,1890,1332,1195,1716
2540 DATA1506,1210,1875,1898,1270,1300,1749,1934,1175,1147,1946,1980
3000 REM SKRIV STOPIT
3001 POKE53280,0:POKE53281,0:PRINT"[CLR]"
3002 PRINT"[QQQQQQQQQQRRRRRRRRRRRR]";
3005 PRINT"[...]";
3006 REM S
3007 PRINT"[...]";:PRINT"[...]";:FORL=1TO8:PRINT"[...]";:NEXT:PRINT"[...]";
3008 GOSUB3050:REM T
3009 PRINT"[...]";
3010 GOSUB3050:REM O
3011 PRINT"[...]";
3012 GOSUB3050:REM P
3013 PRINT"[...]";"[...]";
3014 GOSUB3050:REM I
3015 PRINT"[QQQQQQQQ]";
3016 PRINT"[...]";:PRINT"[...]";:FORL=1TO8:PRINT"[...]";:NEXT:PRINT"[...]";
3017 GOSUB3050:REM T
3018 PRINT"[QQQQQQQQQQQQQQQQQQQQQQRRRRRRRRRRRRRRRR] PRESS ANY KEY "
3048 POKE646,RND(1)*15:FORT=1TO300:NEXT:GOTO3002
3049 END
3050 FORT=1TO150:NEXT
3055 IFPEEK(198)=1THEN3100:REM KEY PRESSED
3060 RETURN
3100 REM **** RITA RAM *****
3110 POKE53272,23:REM LOWER CASE
3120 GOSUB3150:GOSUB3180:GOTO3200
3150 PRINT"[CLR]┌──────────────────────────────────────┐";
3160 FORL=1TO22:PRINT"│"SPC(38)"│";:NEXT
3170 PRINT"└──────────────────────────────────────┘[S]":RETURN
3180 PRINT"[QQQQQRRRRR]1. [R]NSTRUCTIONS":PRINT"[RRRRR]2. [R]LAY":RETURN
3200 REM INSTRUKTION ELLER SPEL
3220 GETA$:IFA$="1"THEN3250
3230 IFA$="2"THEN3500
3240 GOTO3220
3250 REM ***** INSTRUKTIONER *****
3255 GOSUB 3150:REM RADERA
3260 PRINT"[QQQRRRR]TOP THE BALL WITH"
3270 PRINT"[RRR]ANY KEY"
3275 PRINT
3280 PRINTTAB(3)"[S]CORE POINTS BY"
3290 PRINTTAB(3)"[S]STOPPING THE BALL"
3300 PRINT"[RRR]ON THE NUMBERS.       "
3310 PRINT"[RRRR]EEP THE BALL WITHIN"
3320 PRINT"[RRR]THE FRAME."
3330 PRINT"[QRRRR]HIT ANY KEY TO PLAY."
3400 REM HIT KEY TO CONTINUE
3420 POKE198,0:WAIT198,1
3500 REM TILL SPELINITIERING
3501 GOTO70
7900 REM *** READ MUSIK ****
7920 PRINT"[CLR]":POKE53280,0:POKE53281,0:S=1024:CM=55296
7930 FORL=1TO36:READA:NEXT
7950 FORL=49152TO49913:READA:POKEL,A
7960 POKES,A:POKECM,RND(1)*15:S=S+1:CM=CM+1:NEXT
7970 SYS49547:PRINT"[CLR]":RETURN
7990 REM ***** MUSIKDATA *****
7992 REM FROM 49152 TO 49900
7995 REM
8000 DATA 0,32,32,0,219,40,39
8001 DATA 0,40,40,0,3,162,0
8002 DATA 232,236,11,192,240,92,254
8003 DATA 4,192,189,4,192,221,7
8004 DATA 192,208,239,224,1,208,6
8005 DATA 169,0,133,253,240,14,224
8006 DATA 2,208,6,169,7,133,253
8007 DATA 208,4,169,14,133,253,189
8008 DATA 0,192,41,254,160,4,145
8009 DATA 253,32,143,192,201,0,240
8010 DATA 44,201,255,208,3,32,209
8011 DATA 192,157,7,192,169,0,157
8012 DATA 4,192,32,143,192,160,0
8013 DATA 145,253,32,143,192,160,1
8014 DATA 145,253,189,0,192,9,1
8015 DATA 160,4,145,253,76,14,192
8016 DATA 76,49,234,160,0,169,0
8017 DATA 145,253,200,192,24,208,249
8018 DATA 169,15,145,253,169,49,141
```

▶

```
8019 DATA 20,3,169,234,141,21,3
8020 DATA 76,49,234,172,4,192,192
8021 DATA 0,208,2,230,252,177,251
8022 DATA 204,202,192,208,46,164,252
8023 DATA 204,203,192,208,39,172,204
8024 DATA 192,185,219,194,141,4,192
8025 DATA 200,185,219,194,133,252,200
8026 DATA 185,219,194,141,202,192,200
8027 DATA 185,219,194,141,203,192,200
8028 DATA 140,204,192,76,143,192,0
8029 DATA 0,18,238,4,192,96,32
8030 DATA 143,192,157,0,192,160,4
8031 DATA 145,253,41,64,240,14,32
8032 DATA 143,192,160,2,145,253,32
8033 DATA 143,192,160,3,145,253,32
8034 DATA 143,192,160,5,145,253,32
8035 DATA 143,192,160,6,145,253,32
8036 DATA 143,192,168,41,7,240,26
8037 DATA 152,160,23,145,253,32,143
8038 DATA 192,160,21,145,253,32,143
8039 DATA 192,160,22,145,253,32,143
8040 DATA 192,160,24,145,253,32,143
8041 DATA 192,96,169,0,141,204,192
8042 DATA 169,219,141,170,192,141,177
8043 DATA 192,141,183,192,141,190,192
8044 DATA 141,82,193,141,89,193,169
8045 DATA 194,141,171,192,141,178,192
8046 DATA 141,184,192,141,191,192,141
8047 DATA 83,193,141,90,193,172,204
8048 DATA 192,185,219,194,141,202,192
8049 DATA 200,185,219,194,141,203,192
8050 DATA 200,140,204,192,32,143,192
8051 DATA 141,11,192,162,1,169,0
8052 DATA 157,4,192,169,1,157,7
8053 DATA 192,232,236,11,192,208,240
8054 DATA 169,212,133,254,120,169,12
8055 DATA 141,20,3,169,192,141,21
8056 DATA 3,88,96,169,166,141,4
8057 DATA 192,169,193,133,252,169,219
8058 DATA 141,39,193,169,194,141,59
8059 DATA 193,169,15,141,24,212,76
8060 DATA 33,193,3,255,32,10,0
8061 DATA 0,5,0,0,255,32,10
8062 DATA 0,0,5,0,0,7,154
8063 DATA 21,28,37,17,7,227,22
8064 DATA 14,177,25,7,214,28,28
8065 DATA 107,14,7,94,32,14,75
8066 DATA 34,7,126,38,28,114,11
8067 DATA 7,52,43,14,198,45,14
8068 DATA 126,38,28,216,12,14,94
8069 DATA 32,14,94,32,28,177,25
8070 DATA 14,177,25,14,177,25,28
8071 DATA 47,16,14,227,22,7,154
8072 DATA 21,28,37,17,7,227,22
8073 DATA 14,177,25,7,214,28,28
8074 DATA 107,14,7,94,32,14,75
8075 DATA 34,7,126,38,28,114,11
8076 DATA 7,52,43,14,198,45,14
8077 DATA 126,38,28,216,12,14,94
8078 DATA 32,14,177,25,28,177,25
8079 DATA 14,94,32,28,75,34,28
8080 DATA 37,17,7,52,43,7,37
8081 DATA 17,7,198,45,7,63,19
8082 DATA 14,97,51,14,154,21,7
8083 DATA 52,43,7,37,17,7,198
8084 DATA 45,7,63,19,14,97,51
8085 DATA 14,154,21,7,52,43,7
8086 DATA 37,17,7,198,45,7,63
8087 DATA 19,14,97,51,14,154,21
8088 DATA 14,172,57,28,114,11,14
8089 DATA 198,45,14,198,45,28,126
8090 DATA 38,14,126,38,14,126,38
8091 DATA 28,63,19,14,214,28,7
8092 DATA 126,38,7,47,16,7,52
8093 DATA 43,7,37,17,14,198,45
8094 DATA 14,63,19,7,126,38,7
8095 DATA 47,16,7,52,43,7,37
8096 DATA 17,14,198,45,14,63,19
8097 DATA 7,126,38,7,47,16,7
8098 DATA 52,43,7,37,17,14,198
8099 DATA 45,14,63,19,14,97,51
8100 DATA 28,154,21,14,52,43,14
8101 DATA 52,43,28,37,17,14,75
8102 DATA 34,14,75,34,28,37,17
8103 DATA 14,75,34,40,75,34,40
8104 DATA 37,17,0,50,194,183,193
8105 DATA 212,194,183,193,212,194,183
8106 DATA 193,44,194,212,194,0,0
8107 DATA120,169,49,141,20,3,169,234,141,21,3,88,96

READY.
```



□

GRATISANNONSER

SÄLJES

TV 64 PÅ KASSETT
Detta är två program i ett till VIC 64. Man kan räkna ut volymen av en boll, kon m m och omvandla Celsius till Fahrenheit m m. Skicka mig 35 kr + band så får du programmet, Matts Rydberg, Odensvivägen 12A, 730 30 Kolsva.

VIC 20 SÄLJES
VIC 20, bandspelare, expansionsenhet VIC 1020, super expander +3k, 16k, prog-aid, forth, joystick, böcker, 80-tal spel- och nyttoprogram. Säljes helt eller i delar. Ring 0150-188 65.

SKRIVARE SÄLJES
Seikosha GP 100 VC till VIC64 säljes p g a byte, 1750:–. 0978-203 27, Fredrik.

VIC 20 SÄLJES
VIC 20 med bandspelare + 16k, 3k ram + 3 st böcker bla. "Från spel till Basic" + ca 100 spel- + nytto-program säljes för endast 3000 kr ring fort 031-93 16 19. Prata med Daniel. Jag säljer även VIC 64 spel, har massor! Skriv till Jarl Bennis, Transformatorv. 7, 430 41 Kullavik.

SÄLJES TILL VIC 64
Programming kit. 1 från Timeworks. Du lär dig med detta program göra egna spel. Nypris: 295:–. Nu endast 150:– för originalkassett med manual. Erik Bertell, Ringvägen 23, 824 00 Hudiksvall, tele: 0650-161 74.

GLOSFÖRHÖR FÖR VIC 64
Funktioner t ex utskrift av glosor, förhör, ändra, spara, hämta glosor. Pris: 50 kr + kassett el. diskett. Skriv till Johan Harrysson, Båtvägen 5, 590 61 Vreta Kloster, för beställning eller mer information.

VIC 20 tillbehör
Forth + 3k 275 kr. Böckerna Zap Pow Boom och VIC Innovate computing 60 kr/st. eller 100 kr för båda. Originalspelet Andes Attack 50 kr. Miguel Berg, Fridhemsgatan 110 C, 852 43 Sundsvall.

DATATIDNINGAR
Mikrodatorn 9, 10 -83, 1–6 -84 Persondatorn 6, 7, 8 -84 Min Hemdator 1, 3, 4 -83, 1–5 -84. Allt om hemdatorer 3–5 -83, 1–8 -84. Säljes för 200:– + porto. Ring 040-12 89 12.

VIC 20 SPEL SÄLJES
3 st för 50 kr. Missile, Caterpillar, Animals, 3D-Man + 3k (Js), Fanbs (ormen), Frog Run, Poker + 3k, Yatzy + 3k, Lotto med Rättning + 3k, Ladders + 8k. Vilka vill du ha? Tore Gullstrand, Norrgårdsv. 127, 184 00 Åkersberga.

VIC 20 SÄLJES
VIC 20 med bandspelare, instruktionsbok samt ca 200 spel och nyttoprogram säljes för 1100 kr. Tel 08-47 12 97 efter kl. 18.00.

SPEL TILL VIC 20
5 spel på kassett säljes för 150 kr (obs ej piratkopior), frakt 15:– tillkommer. Vid beställning inom 15 dagar tillkommer 5 extra program. Per Christensen, Box 94, 574 00 Vetlanda.

VIC 64
Nyttoprogram säljes till högst bjudande. Teledata Text 64 m m. Tel: 018-25 03 55 kl. 16–18.

VIC 64 PROGRAM SÄLJES
Zeppelin, Forth Apocalypse, Survivor, Mule, Zaxxon, Astro Chase, Solo Flight, Choplifter, 125 kr/st. Simons basic 250 kr, Graf 64 200kr. Observera allt är original! Kan även byta. Anders Gorner, Harrebacken 6, 443 00 Lerum.

NY VIC 64 SÄLJES
4 mån. gammal VIC 64 + bandst + manual + Intr. Basic del 1 + System VIC 64 + div. spel (Cup Final, Blue Max, Pharaos Curse m m 30 st.) allt för 3350 kr 0910-109 32 Peter efter kl. 16.00.

VIC 20 SÄLJES MED
bandstation, 16k ram, ljuspenna, joystick, extra joystickport, 6 program-böcker, en lista på spel, manual Zorgon's Kingdom, manual till Boss, joystickrutin, DataBas manual, 65 spel och nyttoprogram. Pris 2500 kr, kontakta Johan på tel 08-771 27 41!

PROGRAM VIC -64
Simons Basic 400 kr, Forth 64 350 kr, Hesmon 250 kr, tel 031-20 75 69.

SPEL SÄLJES TILL VIC 20
Skriv till Patrik Skoog, Pl.5020, 453 00 Lysekil, så skickar jag prislista (skicka med returporto).

****DAMMSKYDD****
Skydda din dator mot smuts och damm. VIC 20/64 + bandpelare: 40 kr, Disk drive 1541: 40 kr. Även special skydd kan beställas. Tel 0303-776 15, Lars Olsson.

PRINTER SÄLJES
Printer VIC 1525 säljes för 2400 kr. Tel 0500-307 50.

HAR DATORN LÅST SIG?
Skaffa dig en resetknapp för bara 40 kr. Ring Hans 013-703 79.

FORTH TILL CBM 64
Forth cartaridge-modul + PET forth manual säljes för 600:– (nypris 770:–), ring 060-927 40 Roland.

VIC 20 SÄLJES
Bandspelare, extra handböcker, en årgång VIC rapport, ca 50 spel t ex Arcadia, Megagalatic, Llamas + många fler bra spel, pris 1400 eller högst erbjudande. Hör av er till Tommy, 0321-810 71 efter 17.00.

VIC 20 SÄLJES MED
många spel bl a. Superlander, Crazy, Kong, Hoppit, Arcadia, Vic, Panic + programmerings böcker + joystick, pris 1300 kr, ring till Gunnar tel 0533-190 56.

VIC 20 SÄLJES
VIC 20, bandspelare, maskinkodsspel, 2 cartridgespel, +3k, Programmers Aid, 2 joysticks, 1 originalkassett, böcker, tidningar för bara 3995 kr. Kontakta Fredrik 0383-175 36 slå till direkt! OBS! Många spel!

LOST.? NEVER FEAR..!
Twas is here? Your CBM 64 Adv Guide. Lösningar till: Adv inom (parentes) end.? tips. Scott Adams fem första. (The Hulk) (Snow Ball) (Col Adv) (Twin King Val) (The Hobbit) TWAS: The Wheel adv service OBS! + Caztec tomb, ring Mattias 013-15 21 20.

KOMPLETT 64 SYSTEM
CBM 64, Floppy, 1526-Printer, Joystick, 1701-Monitor, ca 60 diskar med program och spel, böcker, tidningar m m. m m. Nypris: över 20 000:–, bortslumpas för 15 000:– eventuellt även i delar. Ring Mats tel 021- 33 27 30 nu!

SÄLJES
64:a med 1541 floppy, Calc Result Advance, Mon 64, Hesmon, 60 disketter m 500 program, 3 joysticks och en massa dokumentation. Nypris 12 500:– exkl. programvara, nu 9000:– eller högstbjudande. Kontakta Magnus tel 021- 33 01 03.

VIC 64 PROGRAM SÄLJES
Säljer bl a Superbase 500 kr (nypris 995 kr), Beach head, Summer Games 200 kr, Hover Bovver 30 kr. OBS alla program är original. Ring Andre på tel 08-85 66 49.

ORIGINALSPEL TILL VIC 64
Siren City 39 kr, Frogger 64 39 kr, ring Sven på tel 08-773 24 36.

PHILIPS TV-SPEL G7000
Med fyra kassetter säljes till det förmånliga priset 500:–, billigare än ett enda spel till din dator! Ring 0380-113 83 för vidare information.

SÄLJES OCH BYTES
Spel till VIC 64 bytes och säljes. Program till Simon's Basic köpes eller bytes. Även nyttoprogram säljes och bytes. Skriv till Keivan Kechmiri, Gösvägen 29, 663 02 Hammarö eller ring 054-247 14.

BANDMINNE
Commodore datadiskett modell C2N 275:–, Bertil Nilbo 042-22 24 36.

TRASIG VIC?
Har du en trasig VIC eller en annan hemdator. Ring mig, jag kan nog hjälpa dig med den. Leif 08-740 17 11 mellan 18–21.

ASSEMBLER & MONITOR
till VIC 64 i 800 kr klassen. 195:– inkl. allt. Sätt in 20:– på Pg 17 52 00–5 så får Du manualen. Den dras av vid köpet. BIG FORTH med assembler och fullskärmseditor 195:– Tel 0515-103 00 ROJTEC, Box 2, 521 02 Falköping.

BYTES

CARTRIDGE BYTES
Cartridge-spelet Lemans bytes mot annat spel eller nyttoprogram på cartridge. Ev säljes. Per Grundström, Cederrothsväg 12, 663 00 Skoghall, tel 054-237 35.

64 SPEL BYTES
Jag byter spel och nyttoprogram. Stort sortiment. Tel: 090-13 38 42, Niclas Åström.

COMMODORE 64
Jag önskar byta och sälja spel på kassett (mycket billigt) till VIC 64, Ring eller skriv till Lars Tingvall, Slåttervägen 3, 824 00 Hudiksvall, tel: 0650-167 83 (efter 15.00).

COMMODORE 64
Spel bytes och säljes både på disk och band. Söker spel som Pole Position II, Xavius, Jupiter Mission, Robots of Dawn och Combatlynx, skriv till Tommy Isaksson, Furudalsv. 15, 752 60 Uppsala.

VIC 20 SPEL BYTES
Jag vill byta oexp och exp spel till VIC 20 ev säljes billigt. Ring 0411-332 39 och fråga efter Johan eller skriv till. Johan Adolfsson, Högastensvägen 6, 270 11 Abbekås.

VIC 20 SPEL BYTES
Jag byter oexpanderade spel till VIC 20 t ex Scramble, Amok, Llamasoft o s v. Är du intresserad ring 0300-616 13, fråga efter Anders.

VIC 64 SPEL BYTES
Jag byter mina högklassiga maskinkodsspel mot dina. Tveka inte! Ring därför direkt till Niklas, 0250-168 65.

VIC 64 PROGRAM BYTES
eller säljes. Anders Gorner, Hareb 6, 443 00 Lerum.

DATAKOMPIS
Datakompis med VIC 20 i Norrköpings-trakten sökes för byte av erfarenheter. Ring Robin 011-587 70.

VIC 20 o VIC/CBM 64 BYTES
Skicka 1–4 m-kodspel på kassett så skickar jag dig lika många. Jonny Karlsson, Åkerby 1403, 816 00 Ockelbo.

SPELGALNINGAR
Vill du byta spel? Jag har Hobbit, Congo Bongo, Zeppelin, Hero, Hungry Horace och div andra spel, endast på kassett! Kontakta Magnus Bergqvist, Sparrefeltsgatan 318, 502 57 Borås.

VIC 64 SPEL BYTES
Jag har ca 50 st spel och jag har även nyttoprogram. Jag har bl a Zaxxon, Bruce Lee, Ghostbusters, B.C's Quest for Tires. Ring eller skriv till Sven Lind, Klisätravägaen 41, 138 00 Älta, tel: 08-773 24 36.

BYTES
Jag skule vilja byta Dallas Quest mot något annat roligt spel t ex Gateway To Apshai. Släng dig på telefonen och ring: 08-716 34 85, fråga efter Björn.

HEJSAN, HEJSAN ALLIHOP
Vi är två 64 innehavare som gärna vill byta lite ruggigt häftiga spel vi har bl a Dallas Quest, Raid over Moscow, Sam, Mule och varsin 64 dator. Vi har även 150 andra super häftiga spel! Ring 031-28 26 89, fråga efter spelidioten Niklas.

VIC 64
Jag vill byta till mig Valhalla och/eller Knights of the Desert. Har ca 20 spel t ex Manic Miner, Decathlon, Blue Max, Fort Apocalypse. Tel 0303- 816 95 Per.

COMMODORE
Spel och nyttoprogram bytes. Skicka lista till Krister Svensson, Kvarng 61, 267 00 Bjuv.

VIC 64 SPEL BYTES! VIC 64
The Hobbit och Blue Max och andra spel bytes mot andra spel. Helst i maskinkod. Spelen ska vara på kassett. Ring tel 023-237 76, fråga efter Mattias efter kl 16.00.

KÖPES

KÖPES
8 eller 16k expansionsminne köpes till VIC 20, max 350 kr, Miguel Berg, Fridhemsgatan 110C, 852 43 Sundsvall.

VIC 20
16k byte extraminne önskas köpa för 250 kr. Skriv till Patrik Skoog, Pl 5020, 453 00 Lysekil, snarast!

SE HIT!
Jag köper din flygsimulator till VIC 20, ring 019-13 47 13 efter kl 17.00, fråga efter Mikael.

KÖPES TILL VIC 20
16k minne köpes för högst 200 kr. Skriv till: Patrik Skoog, Pl 5020, 453 00 Lysekil.

SIMON'S* MANUAL KÖPES
Jag köper SIMONS'S BASICS instruktionsbok till ett rimligt pris. Byter gärna spel och nyttoprogram också. Ring 0370-144 37 och fråga efter Henrik.

KÖPES
Valhalla 64, (originalkassett) samt 64 Ref Guide. Skriv till: Dean Tosic, Lokvägen 37, 260 33 Påarp, och ange hur mycket du vill ha!

KÖPES TILL VIC 20
Action och äventyrsspel skicka med lista till: Patrik Skoog, Pl 5020, 453 00 Lysekil.

VIC rapport nummer 3 utkommer den 3 april.

Tipsextradax

Många läsare har efterlyst ett tipsprogram till VIC 20. I nästa nummer finner du ett sådant.

En novell

I nästa nummer kan du läsa om Lars, som blev "biten" av adventurespel.

Dataregister 64 advanced

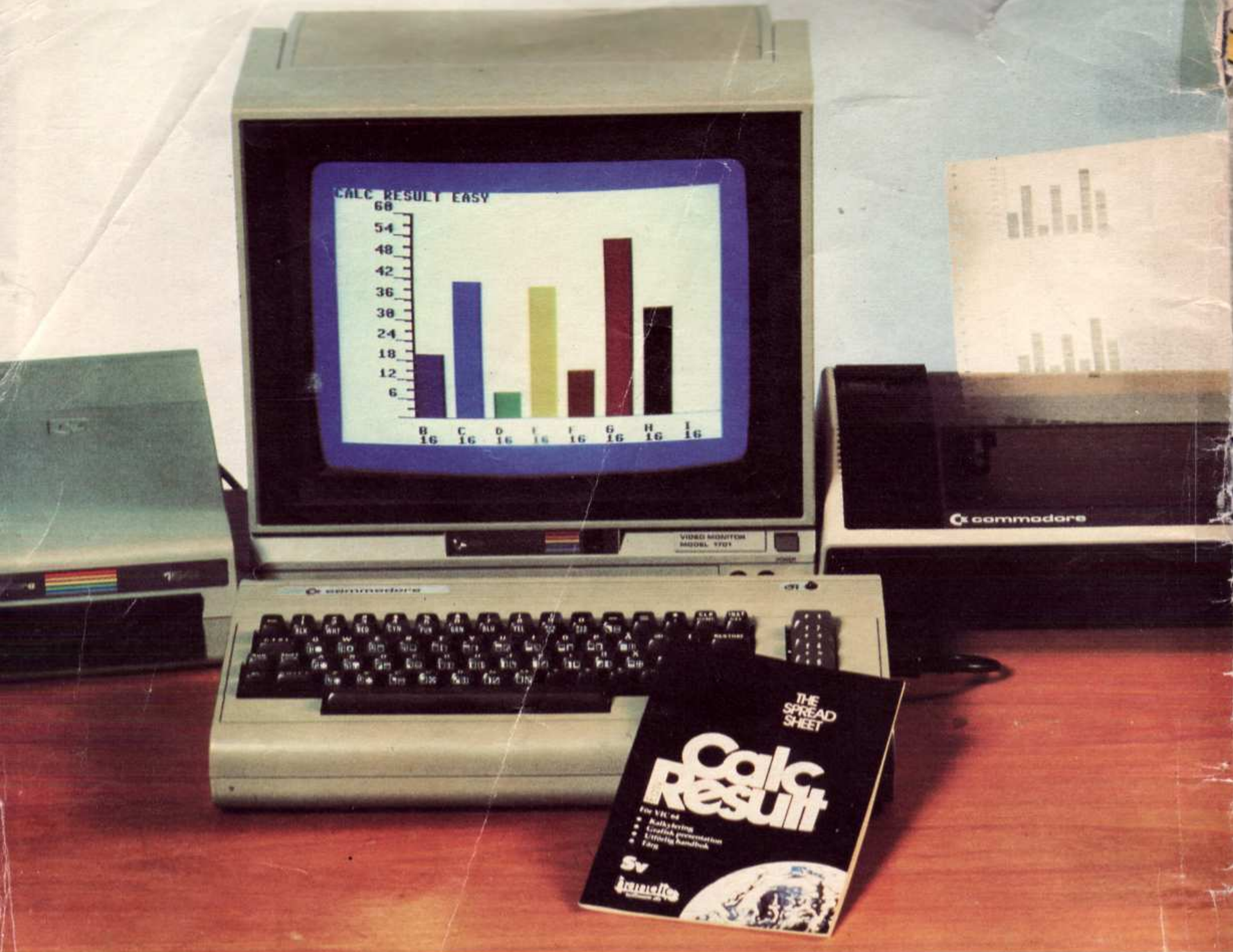
VIC Center i Stockholm kommer i dagarna ut med ett nytt program. Läs rescensionerna om detta i nästa nummer.

Årets datorer utsedda

Vilken är årets hemdator? Svaret får du i VIC rapport nr 3 1985.

Slut på lagret

VIC rapport årg 3 nr 8 och 9 samt nr 1/85 är slut på lagret. Går ej att beställa.



Räkna med VIC 64S

"Pluggar" du in Calc Result i din VIC 64S så får du en kombination som har blivit en världssuccé

Calc Result är ett kalkyleringsprogram, som finns i två versioner.

Calc Result Easy är ett sk "spreadsheet" dvs ett jätteark där du kan bygga upp dina kalkylmodeller, koppla ihop faktorer som är beroende av varandra, göra grafisk presentation av värden, ändra en faktor i en total kalkyl och blixtsnabbt få det nya resultatet. Bygg tex upp deklarationsblanketten och spara in den, så går nästa års deklaration på en kafferast.

Calc Result Advanced arbetar på samma sätt, men har en tredje dimension. Det har 32 sidor att arbeta på, vilket gör det möjligt att på ett företag kalkylera flera avdelningars resultat över hela året. Varje avdelning lägger du på en egen sida i Calc Result Advanced. När samtliga avdelningar är färdigräknade var för sig summerar du helt enkelt resultatet till sista sidan och får omedelbart en total sammanställning.

Kalkylering är en del av vad VIC 64S kan. I foldern "VIC till vardags" får du mer information om vad VIC kan göra för dig i vardagslivet.

handic electronic ab

Box 1063, 436 00 Askim/Göteborg, Tel. 031/28 97 90
– ett företag i Datatronicgruppen –

Jag vill ha foldern "VIC till vardags"

Namn ____________________

Adress ____________________

Postnr/Ort ____________________

Sänd in kupongen till handic electronic, Box 1063, 436 00 Askim.